Pang-Ning Tan
Sanjay Chawla
Chin Kuan Ho
James Bailey (Eds.)

LNAI 7302

Advances in Knowledge Discovery and Data Mining

16th Pacific-Asia Conference, PAKDD 2012
Kuala Lumpur, Malaysia, May/June 2012
Proceedings, Part II

2 Part II

Springer

Lecture Notes in Artificial Intelligence 7302

Subseries of Lecture Notes in Computer Science

Pang-Ning Tan Sanjay Chawla
Chin Kuan Ho James Bailey (Eds.)

Advances in Knowledge Discovery and Data Mining

16th Pacific-Asia Conference, PAKDD 2012
Kuala Lumpur, Malaysia, May 29 – June 1, 2012
Proceedings, Part II

Springer

Volume Editors

Pang-Ning Tan
Michigan State University, Department of Computer Science and Engineering
428 S. Shaw Lane, 48824-1226 East Lansing, MI, USA
E-mail: ptan@cse.msu.edu

Sanjay Chawla
University of Sydney, School of Information Technologies
1 Cleveland St., 2006 Sydney, NSW, Australia
E-mail: sanjay.chawla@sydney.edu.au

Chin Kuan Ho
Multimedia University, Faculty of Computing and Informatics
Jalan Multimedia, 63100 Cyberjaya, Selangor, Malaysia
E-mail: ckho@mmu.edu.my

James Bailey
The University of Melbourne, Department of Computing and Information Systems
111 Barry Street, 3053 Melbourne, VIC, Australia
E-mail: baileyj@unimelb.edu.au

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-30219-0 e-ISBN 978-3-642-30220-6
DOI 10.1007/978-3-642-30220-6
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012937031

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

LNCS Sublibrary: SL 7 – Artificial Intelligence

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

Printed on acid-free paper

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

Preface

PAKDD 2012 was the 16th conference of the Pacific Asia Conference series on Knowledge Discovery and Data Mining. For the first time, the conference was held in Malaysia, which has a vibrant economy and an aspiration to transform itself into a knowledge-based society. Malaysians are also known to be very active in social media such as Facebook and Twitter. Many private companies and government agencies in Malaysia are already adopting database and data warehousing systems, which over time will accumulate massive amounts of data waiting to be mined. Having PAKDD 2012 organized in Malaysia was therefore very timely as it created a good opportunity for the local data professionals to acquire cutting-edge knowledge in the field through the conference talks, tutorials and workshops.

The PAKDD conference series is a meeting place for both university researchers and data professionals to share the latest research results. The PAKDD 2012 call for papers attracted a total of 241 submissions from 32 countries in all six continents (Asia, Europe, Africa, North America, South America, and Australasia), of which 20 (8.3%) were accepted for full presentation and 66 (27.4%) were accepted for short presentation. Each submitted paper underwent a rigorous double-blind review process and was assigned to at least four Program Committee (PC) members. Every paper was reviewed by at least three PC members, with nearly two-thirds of them receiving four reviews or more. One of the changes in the review process this year was the adoption of a two-tier approach, in which a senior PC member was appointed to oversee the reviews for each paper. In the case where there was significant divergence in the review ratings, the senior PC members also initiated a discussion phase before providing the Program Co-chairs with their final recommendation. The Program Co-chairs went through each of the senior PC members' recommendations, as well as the submitted papers and reviews, to come up with the final selection. We thank all reviewers (Senior PC, PC and external invitees) for their efforts in reviewing the papers in a timely fashion (altogether, more than 94% of the reviews were completed by the time the notification was sent). Without their hard work, we would not have been able to see such a high-quality program.

The three-day conference program included three keynote talks by world-renowned data mining experts, namely, Chandrakant D. Patel from HP Labs (*Joules of Available Energy as the Global Currency: The Role of Knowledge Discovery and Data Mining*); Charles Elkan from the University of California at San Diego (*Learning to Make Predictions in Networks*); and Ian Witten from the University of Waikato (*Semantic Document Representation: Do It with Wikification*). The program also included four workshops, three tutorials, a doctoral symposium, and several paper sessions. Other than these intellectually inspiring events, participants of PAKDD 2012 were able to enjoy several social events

throughout the conference. These included a welcome reception on day one, a banquet on day two and a free city tour on day three. Finally, PAKDD 2012 organized a data mining competition for those who wanted to lay their hands on mining some real-world datasets.

Putting a conference together with a scale like PAKDD 2012 requires tremendous efforts from the organizing team as well as financial support from the sponsors. We thank Takashi Washio, Jun Luo and Hui Xiong for organizing the workshops and tutorials, and coordinating with the workshop/tutorial organizers/speakers. We also owe James Bailey a big thank you for preparing the conference proceedings. Finally, we had a great team of Publicity Co-chairs, Local Organization Co-chairs, and helpers. They ensured the conference attracted many local and international participants, and the conference program proceeded smoothly.

We would like to express our gratitude to SAS, AFOSR/AOARD (Air Force Office of Scientific Research/Asian Office of Aerospace Research and Development), MDeC (Multimedia Development Corporation), PIKOM (Computer Industry Association of Malaysia) and other organizations for their generous sponsorhip and support. We also wish to thank the PAKDD Steering Committee for offering the student travel support grant and the grant for the best student paper award(s), and UTAR and MMU for providing the administrative support.

Philip Yu
Ee-Peng Lim
Hong-Tat Ewe
Pang-Ning Tan
Sanjay Chawla
Chin-Kuan Ho

Organization

Organizing Committee

Conference Co-chairs

Philip Yu	University of Illinois at Chicago, USA
Hong-Tat Ewe	Universiti Tunku Abdul Rahman, Malaysia
Ee-Peng Lim	Singapore Management University, Singapore

Program Co-chairs

Pang-Ning Tan	Michigan State University, USA
Sanjay Chawla	The University of Sydney, Australia
Chin-Kuan Ho	Multimedia University, Malaysia

Workshop Co-chairs

Takashi Washio	Osaka University, Japan
Jun Luo	Shenzhen Institute of Advanced Technology, China

Tutorial Co-chair

Hui Xiong	Rutgers University, USA

Local Organization Co-chairs

Victor Tan	Universiti Tunku Abdul Rahman, Malaysia
Wen-Cheong Chin	Multimedia University, Malaysia
Soung-Yue Liew	Universiti Tunku Abdul Rahman, Malaysia

Publicity Co-chairs

Rui Kuang	University of Minnesota, USA
Ming Li	Nanjing University, China
Myra Spiliopoulou	University of Magdeburg, Germany

Publication Chair

James Bailey	University of Melbourne, Australia

Local Arrangements Committee

Soung Yue Liew (Co-chair)	Victor Tan (Co-chair)
Wen Cheong Chin (Co-chair)	Kok Why Ng (Co-chair)
Nadim Jahangir	Choo Yee Ting
Chee Onn Wong	Chiung Ching Ho
Chong Pei Fen	Hau Lee Tong
Timothy Yap	James Ooi
Kok Leong Chan	Yong Haur Tay
Azurawati	Chian Wen Too
Khong Leng Lim	Mariam
Michelle	Meei Hao Hoo
Kean Vee Sor	Priya
Madhavan	Simon Lau
Chin Chwee Wong	Swee Ling Chean

Steering Committee

Co-chairs

Graham Williams	Australian National University, Australia
Tu Bao Ho	Japan Advanced Institute of Science and Technology, Japan

Life Members

Hiroshi Motoda	AFOSR/AOARD and Osaka University, Japan
Rao Kotagiri	University of Melbourne, Australia
Ning Zhong	Maebashi Institute of Technology, Japan
Masaru Kitsuregawa	Tokyo University, Japan
David Cheung	University of Hong Kong, China
Graham Williams	Australian National University, Australia
Ming-Syan Chen	National Taiwan University, Taiwan, ROC

Members

Huan Liu	Arizona State University, USA
Kyu-Young Whang	Korea Advanced Institute of Science and Technology, Korea
Chengqi Zhang	University of Technology Sydney, Australia
Tu Bao Ho	Japan Advanced Institute of Science and Technology, Japan
Ee-Peng Lim	Singapore Management University, Singapore
Jaideep Srivastava	University of Minnesota, USA
Zhi-Hua Zhou	Nanjing University, China
Takashi Washio	Institute of Scientific and Industrial Research, Osaka University
Thanaruk Theeramunkong	Thammasat University, Thailand

P. Krishna Reddy	International Institute of Information Technology, Hyderabad (IIIT-H), India
Joshua Z. Huang	Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China

Senior Program Committee

Anirban Dasgupta	Yahoo! Research Silicon Valley, USA
Arno Siebes	Universiteit Utrecht, The Netherlands
Bart Goethals	University of Antwerp, Belgium
Bernhard Pfahringer	The University of Waikato, New Zealand
Dacheng Tao	Nanyang Technological University, Singapore
Ee-Peng Lim	Singapore Management University, Singapore
Haixun Wang	Microsoft Research Asia, China
Hisashi Kashima	University of Tokyo, Japan
Jeffrey Xu Yu	The Chinese University of Hong Kong, Hong Kong
Jian Pei	Simon Fraser University, Canada
Jianyong Wang	Tsinghua University, China
Jiuyong Li	University of South Australia, Australia
Kyuseok Shim	Seoul National University, Korea
Masashi Sugiyama	Tokyo Institute of Technology, Japan
Ng Wee Keong	Nanyang Technological University, Singapore
Nitesh V. Chawla	University of Notre Dame, USA
Osmar R. Zaiane	University of Alberta, Canada
Panagiotis Karras	Rutgers University, USA
Peter Christen	The Australian National University, Australia
Sameep Mehta	IBM Research, India
Sanjay Ranka	University of Florida, USA
Shivani Agarwal	Indian Institute of Science, India
Wei Wang	University of North Carolina at Chapel Hill, USA
Yu Zheng	Microsoft Research Asia, China

Program Committee

Aditya Krishna Menon	University of California, USA
Aixin Sun	Nanyang Technological University, Singapore
Akihiro Inokuchi	Osaka University, Japan
Albrecht Zimmerman	Katholieke Universiteit Leuven, Belgium
Alexandre Termier	Université Joseph Fourier, France
Alfredo Cuzzocrea	ICAR-CNR and University of Calabria, Italy
Amol Ghoting	IBM T.J. Watson Research Center, USA
Andreas Hotho	University of Kassel, Germany
Andrzej Skowron	University of Warsaw, Poland
Annalisa Appice	Università degli Studi di Bari, Italy

Anne Denton	North Dakota State University, USA
Anne Laurent	Montpellier 2 University, France
Aoying Zhou	East China Normal University, Shanghai, China
Arbee Chen	National Chengchi University, Taiwan
Aristides Gionis	Yahoo! Research, Spain
Aryya Gangopadhyay	University of Maryland, USA
Atsuhiro Takasu	National Institute of Informatics, Japan
Atsuyoshi Nakamura	Hokkaido University, Japan
Benjamin C.M. Fung	Concordia University, Canada
Bettina Berendt	Katholieke Universiteit Leuven, Belgium
Bo Zhang	Tsinghua University, China
Bradley Malin	Vanderbilt University, USA
Bruno Cremilleux	Université de Caen, France
Chandan Reddy	Wayne State University, USA
Chang-Tien Lu	Virginia Polytechnic Institute and State University, USA
Charles Ling	The University of Western Ontario, Canada
Chengkai Li	The University of Texas at Arlington, USA
Chengqi Zhang	University of Technology, Australia
Chiranjib Bhattachar	Indian Institute of Science, India
Choochart Haruechaiy	National Electronics and Computer Technology Center (NECTEC), Thailand
Chotirat Ratanamatan	Chulalongkorn University, Thailand
Chunsheng Yang	Institute for Information Technology, Canada
Clement Yu	University of Illinois at Chicago, USA
Daisuke Ikeda	Kyshu University, Japan
Dan Simovici	University of Massachusetts Boston, USA
Dao-Qing Dai	Sun Yat-Sen University, China
Daoqiang Zhang	Nanjing University of Aeronautics and Astronautics, China
David Albrecht	Monash University, Australia
David Taniar	Monash University, Australia
David Lo	Singapore Management University, Singapore
David F. Gleich	Purdue University, USA
Davood Rafiei	University of Alberta, Canada
Deept Kumar	Virginia Polytechnic Institute and State University, USA
Dejing Dou	University of Oregon, USA
Di Wu	Polytechnic Institute of NYU, USA
Diane Cook	Washington State University, USA
Diansheng Guo	University of South Carolina, USA
Dragan Gamberger	Rudjer Boskovic Institute, Croatia
Du Zhang	California State University, USA
Efstratios Gallopoulos	University of Patras, Greece
Elena Baralis	Politecnico di Torino, Italy

Eyke Huellermeier	University of Marburg, Germany
Fabrizio Silvestri	Istituto di Scienza e Tecnologie dell'Informazione (ISTI), Italy
Feifei Li	Florida State University, USA
Florent Masseglia	INRIA, France
Fosca Giannotti	Università di Pisa, Italy
Francesco Bonchi	Yahoo! Research, Spain
Frans Coenen	University of Liverpool, UK
Gang Li	Deakin University, Australia
Gao Cong	Nanyang Technological University, Singapore
George Karypis	University of Minnesota, USA
Giuseppe Manco	Università della Calabria, Italy
Graham Williams	Australian Taxation Office, Australia
Hady Lauw	Institute for Infocomm Research, Singapore
Haibin Cheng	Yahoo! Labs, USA
Haimonti Dutta	Columbia University, USA
Hanghang Tong	IBM T.J. Watson Research Center, USA
Harry Zhang	University of New Brunswick, Canada
Hassab Elgawi Osman	University of Tokyo, Japan
Hideo Bannai	Kyshu University, Japan
Hiroyuki Kawano	Nanzan University, Japan
Hong Cheng	The Chinese University of Hong Kong, Hong Kong
Hua Lu	Aalborg University, Denmark
Huan Liu	Arizona State University, USA
Hui Wang	University of Ulster, UK
Huidong Jin	Chinese University of Hong Kong, Hong Kong
Ioannis Androulakis	Rutgers University, USA
Irena Koprinska	University of Sydney, Australia
Ivor Tsang	The Hong Kong University of Science and Technology, Hong Kong
Jaakko Hollmen	Aalto University, Finland
James Caverlee	Texas A&M University, USA
Jason Wang	New Jersey's Science and Technology University, USA
Jean-Francois Boulicaut	Université de Lyon, France
Jean-Marc Petit	Université de Lyon, France
Jeffrey Ullman	Stanford University, USA
Jialie Shen	Singapore Management University, Singapore
Jian Yin	Sun Yat-Sen University, China
Jieping Ye	Arizona State University, USA
Jinze Liu	University of Kentucky, USA
John Keane	The University of Manchester, UK
Josep Domingo-Ferrer	Universitat Rovira i Virgili, Spain
Junbin Gao	Charles Sturt University, Australia
Junping Zhang	Fudan University, China

Kamalika Das	NASA Ames Research Center, USA
Kanishka Bhaduri	NASA, USA
Keith Marsolo	Cincinnati Children's Hospital Medical Center, USA
Keith Chan	The Hong Kong Polytechnic University, Hong Kong
Kennichi Yoshida	University of Tsukuba, Japan
Kitsana Waiyamai	Kasetsart University, Thailand
Konstantinos Kalpakis	University of Maryland Baltimore County, USA
Kouzou Ohara	Aoyama-Gakuin University, Japan
Krishnamoorthy Sivakumar	Washington State University, USA
Kun Liu	Yahoo! Labs, USA
Kuo-Wei Hsu	National Chengchi University, Taiwan
Larry Hall	University of South Florida, USA
Larry Holder	Washington State University, USA
Latifur Khan	University of Texas at Dallas, USA
Liang Wang	NLPR, Institute of Automation Chinese Academy of Science, China
Lim Chee Peng	Universiti Sains Malaysia, Malaysia
Lisa Singh	Georgetown University, USA
Maguelonne Teisseire	Maison de la Teledetection, France
Manabu Okumura	Japan Advanced Institute of Science and Technology, Japan
Marco Maggini	Università degli Studi di Siena, Italy
Marian Vajtersic	University of Salzburg, Austria
Marut Buranarach	National Electronics and Computer Technology Center, Thailand
Mary Elaine Califf	Illinois State University, USA
Marzena Kryszkiewicz	Warsaw University of Technology, Poland
Masayuki Numao	Osaka University, Japan
Masoud Makrehchi	University of Waterloo, Canada
Matjaz Gams	J. Stefan Institute, Slovenia
Mengjie Zhang	Victoria University of Wellington, New Zealand
Michael Hahsler	Southern Methodist University, USA
Michael Bruckner	University of Potsdam, Germany
Michalis Vazirgianni	INRIA/FUTURS, France
Min Song	New Jersey Institute of Technology, USA
Min Yao	Zhejiang University, China
Ming-Syan Chen	National Taiwan University, Taiwan
Mingli Song	Zhejiang University, China
Mirco Nanni	Istituto di Scienza e Tecnologie dell'Informazione (ISTI), Italy
Murali Mani	Worcester Polytechnic Institute, USA
Murat Kantarcioglu	University of Texas at Dallas, USA
Nagaraj Kota	Yahoo! Labs, India
Ngoc-Thanh Nguyen	Wroclaw University of Technology, Poland

Olivia Sheng	University of Utah, USA
Pabitra Mitra	Indian Institute of Technology Kharagpur, India
Panagiotis Papadimitriou	Stanford University, USA
Philippe Lenca	Telecom Bretagne, France
Ping Li	Cornell University, USA
Qi Li	Western Kentucky University, USA
Qi He	IBM Research, USA
Qingshan Liu	NLPR, Institute of Automation Chinese Academy of Science, China
Richi Nayak	Queensland University of Technologies, Australia
Robert Hilderman	University of Regina, Canada
Roberto Bayardo	Google, Inc, USA
Rohan Baxter	Australian Taxation Office, Australia
Rui Camacho	Universidade do Porto, Portugal
Ruoming Jin	Kent State University, USA
Sachindra Joshi	IBM Research, India
Sanjay Jain	National University of Singapore, Singapore
Scott Sanner	Australian National University, Australia
See-Kiong Ng	Singapore University of Technology and Design, Singapore
Selcuk Candan	Arizona State University, USA
Shashi Shekhar	University of Minnesota, USA
Shen-Shyang Ho	California Institute of Technology, USA
Sheng Zhong	State University of New York at Buffalo, USA
Shichao Zhang	University of Technology, Australia
Shiguang Shan	Institute of Computing Technology Chinese Academy of Sciences, China
Shoji Hirano	Shimane University, Japan
Shu-Ching Chen	Florida International University, USA
Shuigeng Zhou	Fudan University, China
Shusaku Tsumoto	Shimane University, Japan
Shyam-Kumar Gupta	Indian Institute of Technology, India
Silvia Chiusano	Politecnico di Torino, Italy
Songcan Chen	Nanjing University of Aeronautics and Astronautics, China
Sourav S. Bhowmick	Nanyang Technological University, Singapore
Srikanta Tirthapura	Iowa State University, USA
Srivatsan Laxman	Microsoft Research, India
Stefan Rueping	Fraunhofer IAIS, Germany
Sung-Ho Ha	Kyungpook National University, Korea
Szymon Jaroszewicz	University of Massachusetts Boston, USA
Tadashi Nomoto	National Institute of Japanese Literature, Japan
Takehisa Yairi	University of Tokyo, Japan

Takeshi Fukuda	IBM, Japan
Tamir Tassa	The Open University, Israel
Tao Li	Florida International University, USA
Tapio Elomaa	Tampere University of Technology, Finland
Tetsuya Yoshida	Hokkaido University, Japan
Thepchai Supnithi	National Electronics and Computer Technology Center, Thailand
Thomas Seidl	RWTH Aachen University, Germany
Tom Croonenborghs	Katholieke Hogeschool Kempen, Belgium
Toon Calders	Eindhoven University of Technology, The Netherlands
Toshihiro Kamishima	National Institute of Advanced Industrial Science and Technology, Japan
Toshiro Minami	Kyushu University Library, Japan
Tru Cao	Ho Chi Minh City University of Technology, Vietnam
Tsuyoshi Murata	Tokyo Institute of Technology, Japan
Tu-Bao Ho	Japan Advanced Institute of Science and Technology, Japan
Varun Chandola	Oak Ridge National Laboratory, USA
Vincent S. Tseng	National Cheng Kung University, Taiwan
Vincenzo Piuri	Università degli Studi di Milano, Italy
Vladimir Estivill-Castro	Griffith University, Australia
Wagner Meira	Universidade Federal de Minas Gerais, Brazil
Wai Lam	The Chinese University of Hong Kong, Hong Kong
Walter Kosters	Universiteit Leiden, The Netherlands
Wanpracha Chaovalitw	The State University of New Jersey Rutgers, USA
Wei Fan	IBM T.J. Watson Research Center, USA
Weining Qian	East China Normal University, China
Wen-Chih Peng	National Chiao Tung University, Taiwan
Wilfred Ng	Hong Kong University of Science and Technology, Hong Kong
Woong-Kee Loh	Sungkyul University, South Korea
Xiaofang Zhou	The University of Queensland, Australia
Xiaohua Hu	Drexel University, USA
Xiaohui Liu	Brunel University, UK
Xiaoli Li	Institute for Infocomm Research, Singapore
Xin Wang	University of Calgary, Canada
Xindong Wu	University of Vermont, USA
Xingquan Zhu	Florida Atlantic University, USA
Xintao Wu	University of North Carolina at Charlotte, USA
Xu Sun	Cornell University, USA
Xuan Vinh Nguyen	Monash University, Australia
Xue Li	The University of Queensland, Australia

Xuelong Li	University of London, UK
Xuemin Lin	The University of New South Wales, Australia
Xueyi Wang	Northwest Nazarene University, USA
Yan Liu	IBM Research, USA
Yan Jia	National University of Defense Technology, China
Yang Zhou	Yahoo!, USA
Yang-Sae Moon	Kangwon National University, Korea
Yasuhiko Morimoto	Hiroshima University, Japan
Yi-Dong Shen	Institute of Software, Chinese Academy of Sciences, China
Yi-Ping Chen	La Trobe University, Australia
Yifeng Zeng	Aalborg University, Denmark
Yiu-ming Cheung	Hong Kong Baptist University, Hong Kong
Yong Guan	Iowa State University, USA
Yonghong Peng	University of Bradford, UK
Yue Lu	University of Illinois at Urbana-Champaign, USA
Yun Chi	NEC Laboratories America, Inc., USA
Yunhua Hu	Microsoft Research Asia, China
Zheng Chen	Microsoft Research Asia, China
Zhi-Hua Zhou	Nanjing University, China
Zhiyuan Chen	University of Maryland Baltimore County, USA
Zhongfei Zhang	Binghamton University, USA
Zili Zhang	Deakin University, Australia

External and Invited Reviewers

Aurélie Bertaux	Antonio Bruno
Tianyu Cao	Rui Chen
Zhiyong Cheng	Patricia Lopez Cueva
Jeremiah Deng	Stephen Guo
Raymond Heatherly	Lam Hoang
Peter Karsmakers	Sofiane Lagraa
Stéphane Lallich	Ivan Lee
Peipei Li	Zhao Li
Lin Liu	Corrado Loglisci
Zhenyu Lu	Marc Mertens
Benjamin Négrevergne	Marc Plantevit
Jing Ren	Yelong Sheng
Arnaud Soulet	Vassilios Verykios
Petros Venetis	Guan Wang
Lexing Xie	Yintao Yu
Yan Zhang	

Sponsors


AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
UNITED STATES AIR FORCE


ASIAN OFFICE OF AEROSPACE R&D
BRIDGING SCIENCE & TECHNOLOGY


sas
THE POWER TO KNOW.

Table of Contents – Part II

Pattern Mining: Networks, Graphs, Time-Series and Outlier Detection

Data Manipulation: Pre-processing and Dimension Reduction

Table of Contents – Part I

Supervised Learning: Active, Ensemble, Rare-Class and Online

Unsupervised Learning: Clustering, Probabilistic Modeling

Heterogeneous Ensemble for Feature Drifts in Data Streams

Hai-Long Nguyen[1], Yew-Kwong Woon[2], Wee-Keong Ng[1], and Li Wan[3]

[1] Nanyang Technological University, Singapore
nguy0105@ntu.edu.sg, wkn@acm.org
[2] EADS Innovation Works Singapore
david.woon@eads.net
[3] New York University
wanli@cs.nyu.edu

Abstract. The nature of data streams requires classification algorithms to be real-time, efficient, and able to cope with high-dimensional data that are continuously arriving. It is a known fact that in high-dimensional datasets, not all features are critical for training a classifier. To improve the performance of data stream classification, we propose an algorithm called HEFT-Stream (Heterogeneous Ensemble with Feature drifT for Data Streams) that incorporates feature selection into a heterogeneous ensemble to adapt to different types of concept drifts. As an example of the proposed framework, we first modify the *FCBF* [13] algorithm so that it dynamically update the relevant feature subsets for data streams. Next, a heterogeneous ensemble is constructed based on different online classifiers, including *Online Naive Bayes* and *CVFDT* [5]. Empirical results show that our ensemble classifier outperforms state-of-the-art ensemble classifiers (AWE [21] and OnlineBagging [15]) in terms of accuracy, speed, and scalability. The success of HEFT-Stream opens new research directions in understanding the relationship between feature selection techniques and ensemble learning to achieve better classification performance.

1 Introduction

With rapid technological advancement, many real-life applications, such as stock markets, online stores and sensor networks can produce massive datasets, or *data streams*. To discover knowledge from data streams, scientists have to confront the following challenges: (1) tremendous volumes of data; (2) dynamic changes of the discovered patterns, which is commonly referred to as *concept drifts*; and (3) real-time response. Concept drifts are categorized into two types: gradual drifts with moderate changes and sudden drifts with severe changes. Motivated by the above challenges, there are two common approaches of existing classification models for data streams: online incremental learning and ensemble learning.

Incremental learning trains a single classifier and updates it with newly arrived data. For example: Domingos and Hulten [5] proposed a very fast Hoeffding tree

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 1–12, 2012.

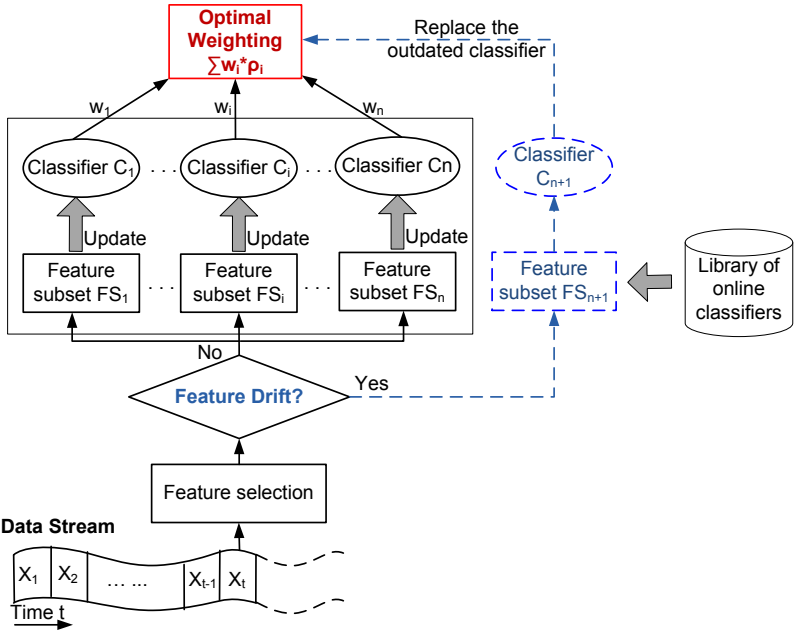


Fig. 1. Ensemble Learning of Feature Drifts

learner (VFDT) for data streams. The VFDT was later extended to CVFDT [12], which can handle the concept drifting streams by constructing alternative nodes and replacing them to the outdated nodes when concept drifts occur. Incremental learning is quite efficient but it cannot adapt to sudden drifts. Ensemble learning, which aims to combine multiple classifiers for boosting classification accuracy, has attracted a lot of research due to its simplicity and good performance. It can manage concept drifts with the following adaptive approaches: (1) using dynamic combiner like a majority vote or weighting combination [18], (2) continuously updating the individual classifiers online [2,15], and (3) changing the ensemble structure by replacing outdated classifiers [16,21]. However, it has high computational complexity as there are many time-consuming processes, eg. generating new classifiers, and updating classifiers.

In this paper, we address the above problems by presenting a novel framework to integrate feature selection techniques and ensemble learning for data streams. To alleviate ensemble updating, we propose a new concept of *"feature drifts"* and use it to optimize the updating process. With a gradual drift, each classifier member is updated in a real-time manner. When a feature drift occurs, which represents a significant change in the underlying distribution of the dataset, we train a new classifier to replace an outdated classifier in the ensemble.

Moreover, feature selection helps to enhance ensemble learning. It not only improves the accuracy of classifier members by selecting the most relevant features

and removing irrelevant and redundant features, but also reduces the complexity of the ensemble significantly as only a small subset of feature space is processed. Finally, we propose a heterogeneous ensemble where different types of classifier members are well selected to maximize the diversity of the ensemble [11,22]. Figure 1 gives an overview of our framework. The ensemble consists of many classifiers, each of which has its own feature subset. If there is a feature drift, the ensemble is updated with a new classifier together with a new feature subset; otherwise, each classifier is updated accordingly. To aggregate the classification results of classifier members, we assign each classier a weight w.r.t its performance. This weighting method is proven to minimize the expected added error of the ensemble.

In summary, the following are contributions of our framework which integrates feature selection and ensemble learning techniques for data streams:

- We enhance ensemble learning with feature selection which helps to lessen computational complexity and increase accuracy.
- We propose the definition of feature drifts and explore relationships between feature drifts and concept drifts.
- We significantly increase the accuracy of the ensemble by designing a heterogeneous ensemble with well-chosen member classifiers and an optimal weighting scheme.

2 Related Work

Ensemble learning, a process to construct accurate classifiers from an ensemble of weak classifiers, has attracted extensive research in the past decade. In general, these methods vary in the way of they construct various classifiers and combine their predictions. The *first* step of constructing a group of classifiers can be differentiated according to the dependencies among classifiers. The independent approach trains classifiers randomly and can be easily parallelized, for example, bagging [3], random subspace [19], and random forest [4]. The dependent approach constructs a new classifier while taking advantage of knowledge obtained during the construction of past classifiers, such as AdaBoost [8], AdaBoost.M2 [7], and Gradient boosting [9]. In the *second* step of combining the classifiers' predictions, majority voting is one intuitive method to choose the dominant decision [3,4,19]. As majority voting cannot guarantee that the voting result will be better than the best individual one, the weighting method is introduced which assigns competent classifiers higher weights, such as performance weighting [7,8,9,21], Naive Bayes weighting [6], and entropy weighting [17].

Ensemble learning can also work well with data streams by effectively tackling the challenges of continuous incoming data and concept drifts [2,15,16,18,21]. In [15], Oza *et al.* employed the Poisson distribution to adapt the traditional bagging and AdaBoost techniques for data streams. Bifet *et al.* [2] proposed an ensemble of Adaptive-Size Hoeffding Trees (ASHT) and used a statistical test to detect concept drifts. Street and Kim [18] proposed a streaming ensemble

of decision trees using majority voting. Wang *et al.* [21] proposed a carefully weighted ensemble for concept-drifting data streams and proved that the ensemble is more accurate than a single classifier trained on the aggregated data of k sequential chunks. In [16], Sattar *et al.* adapted the traditional one-vs-all (OVA) classifiers for data streams where k individual *concept-adapting very fast decision trees* (CVFDT) [12] are learnt and each one is used to distinguish the instances of one class from the instances of all other classes. However, the above algorithms suffer from high complexity and the inability to adapt to different types of concept drifts.

Feature selection is another important research issue as data streams are usually high-dimensional. Feature selection techniques can be classified into three categories: filter, wrapper and embedded models [14]. The filter model evaluates a feature subset by using some independent measure, which only relies on the general characteristics of data. The wrapper model is attached to a learning algorithm and uses its performance to evaluate a feature subset. A hybrid model takes advantage of the above two models. As data streams require real-time responses, we favor the filter approach due to its simplicity and independence to classification models. Moreover, in data streams, the definition of relevant features is dynamic and restricted to a certain period of time. Features that are previously informative may become irrelevant, and previously rejected features may become important features. Thus, dynamic feature selection techniques are required to monitor the evolution of features. Unfortunately, to the best of our knowledge, there is limited research about the relationship between feature selection and ensemble learning, especially for data streams.

In this paper, we will address this gap between feature selection and ensemble learning and propose a novel framework for integrating feature selection and heterogeneous ensembles. Our framework not only adapts to different kinds of concept drifts properly, but also has low complexity due to its dynamic updating scheme and the support of feature selection techniques.

3 Proposed Framework

In this section, we propose a general framework for ensemble learning for data streams. We assume infinite data streams $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_t}, \ldots]$ as input in the framework, where $\mathbf{x_t} = [\mathbf{f_t^1}, \mathbf{f_t^2}, \ldots, \mathbf{f_t^p}]^\mathbf{T}$ is a p-dimensional vector arriving at time t. We assume the data streams have c different class labels. For any data vector $\mathbf{x_t}$, it has a class label $\mathbf{y_t} \in \{\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_c}\}$. Generally when the dimension p is large, there is often only a small set of key features that is critical for building accurate models for classification.

Data streams tend to evolve over time and so do the key features correspondingly. For ease of discussion, we will use the following definitions:

Definition 1. *A **data source** or a concept is defined as set of prior probabilities of the classes and class-conditional probability density function (pdf):*

$$\mathbf{S} = \{(\mathbf{P}(\mathbf{y_1}), \mathbf{P}(\mathbf{X}|\mathbf{y_1})), \ldots, (\mathbf{P}(\mathbf{y_c}), \mathbf{P}(\mathbf{X}|\mathbf{y_c}))\}. \quad (1)$$

Definition 2. *Given data streams X, every instance x_t is generated by a data source or a concept S_t. If all the data is sampled from the same source, i.e. $S_1 = S_2 = \ldots = S_t = S$, we say that the concept is stable. If for any two time points i and j $S_i \neq S_j$, we say that there is a* ***concept drift****.*

Definition 3. *Given a feature space $\mathcal{F}$, at time point t, we can always select the most discriminative subset $\widehat{\mathcal{F}}_t \subseteq \mathcal{F}$. If for any two time points i and j $\widehat{F}_i \neq \widehat{F}_j$, we say that there is a* ***feature drift****.*

Next, we explore the relationship between concept drifts and feature drifts.

Lemma 1. *Concept drifts give rise to feature drifts.*

Proof. Assume that certain feature selection techniques evaluate the discrimination of a feature subset $\mathcal{F}$ at time t by a function:

$$\mathcal{D}(\mathcal{F}_t, t) = \mathcal{D}(P(f^i|y_j), t), \quad f^i \in \mathcal{F}_t \subseteq \{f^1, \ldots, f^p\}, y_j \in \{y_1, \ldots, y_c\} \tag{2}$$

And, there is a feature drift in $[t_i, t_{i+\delta t}]$,

$$\begin{cases} \hat{\mathcal{F}}_t = argmax_{\mathcal{F}_i \subseteq \mathcal{F}} \mathcal{D}(\mathcal{F}_i, t) \\ \mathcal{F}_{t+\delta t} = argmax_{\mathcal{F}_i \subseteq \mathcal{F}} \mathcal{D}(\mathcal{F}_i, t + \delta t) \\ \hat{\mathcal{F}}_t \neq \hat{\mathcal{F}}_{t+\delta t} \end{cases} \tag{3}$$

We can always find a feature $f^a \in \mathcal{F}$ and a class y_b so that $P(f^a|y_b, t) \neq P(f^a|y_b, t + \delta t)$. Else, $P(f^i|y_j, t) = P(f^i|y_j, t + \delta t), \forall (i, j)$, then $\hat{\mathcal{F}}_t = \hat{\mathcal{F}}_{t+\delta t}$. Hence, $P(X|y_b, t) \neq P(X|y_b, t + \delta t) \Rightarrow S_i \neq S_{i+\delta i}$. This denotes a concept drift in the time interval $[t_i, t_{i+\delta t}]$.

Lemma 2. *Concept drifts may not lead to feature drifts.*

Proof. For example, given data streams $\mathbf{X}$ within a time period $[t_i, t_{i+\delta t}]$ we assume the prior probability of a class i and j, $P(y_i)$ and $P(y_j)$, are changed; but their sum $(P(y_i) + P(y_j))$ and other probabilities remain the same. In this scenario, there is a concept drift but no feature drift.

Combining Lemmas 1 and 2, we can conclude that:

Theorem 1. *Feature drifts occur at a slower rate than concept drifts.*

Feature drifts, which are observed in high dimensional data streams, occur no faster than concept drifts. As shown in the overview of our framework Figure 1, we need to modify feature selection techniques to detect feature drifts. The key idea is using feature selection to accelerate ensemble learning and steer its updating process. Moreover, feature selection techniques not only remove irrelevant and redundant features, but also accelerate the learning process by reducing the dimensionality of the data. Thus, the overall performance of the ensemble is improved in terms of accuracy, time and space complexities.

First, we select online classifiers as classifier members as they can be incrementally updated with new data. Then, we construct a heterogeneous ensemble with a capability to adapt to both concept and feature drifts. With gradual drifts, we only need to update the classifier members. With feature drifts, we adjust the ensemble by replacing the outdated classifier with a new one. We further deploy a weighting technique to minimize the cumulative error of the heterogeneous ensemble.

3.1 Feature Selection Block

Feature selection selects a subset of q features from the original p features ($q \leq p$) so that the feature space is optimally reduced. Generally, we expect the feature selection process to remove irrelevant and redundant features. We decide to use FCBF [13] as it is simple, fast and effective. FCBF is a multivariate feature selection method where the class relevance and the dependency between each feature pair are taken into account. Based on information theory, FCBF uses symmetrical uncertainty to calculate dependencies of features and the class relevance. Starting with the full feature set, FCBF heuristically applies a backward selection technique with a sequential search strategy to remove irrelevant and redundant features. The algorithm stops when there are no features left to eliminate.

Symmetrical Uncertainty (SU) uses entropy and conditional entropy values to calculate dependencies of features. If X, Y are random variables, X receives value x_i with probability $P(x_i)$, Y receives value y_j with probability $P(y_j)$; the symmetrical uncertainty between X and Y is:

$$SU(X,Y) = 2\left[\frac{H(X) - H(X|Y)}{H(X) + H(Y)}\right] = 2\left[\frac{I(X,Y)}{H(X) + H(Y)}\right], \tag{4}$$

where $H(X)$ and $H(Y)$ are the entropies of X and Y respectively; $I(X,Y)$ is the mutual information between X and Y; the higher the $SU(X,Y)$ value, the more dependent X and Y are.

We choose to the sliding window version of FCBF so that it has low time and space complexities. Incoming data is stored in a buffer (window) with a predefined size. Next, the matrix of symmetrical uncertainty values is computed to select the most relevant feature subset. The process is performed in a sliding window fashion, and the selected feature subsets are monitored to detect feature drifts. When two consecutive subsets are different, we postulate that a feature drift has occurred.

3.2 Ensemble Block

Heterogeneous Ensemble. When constructing an ensemble learner, the diversity among member classifiers is expected as the key contributor to the accuracy of the ensemble. Furthermore, a heterogeneous ensemble that consists of different classifier types usually attains high diversity [11,23]. Motivated by this

Algorithm 1. Ensemble Learning

Input: A series of infinite streaming data $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_t}, \ldots]$, where $\mathbf{x_t}$ is a p dimensional vector $[f_t^1, f_t^2, \ldots, f_t^p]^T$ with a class label y_t arriving at time t, $y_i \in \{y_1, y_2, \ldots, y_c\}$.
A set of l different classifier types, $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_l\}$.
Output: An heterogenous ensemble $\mathcal{E}$.

```
Initialize the ensemble E with k classifiers of each model in M, denoted as C_1, C_2, ..., C_{k*l}.
while X has more instance do
  if chunk is not full then
    Add x_i to chunk.
  else
    Perform FCBF to get the relevant and non-redundant feature subset φ_i.
    if φ_i ≠ φ_{i-1} then
      Find the best accurate classifier C_best having the smallest aggregated error in the ensemble and get its type.
      Build a new classifier C_new having the same type with C_best, and associated with the feature subset φ_i.
      Remove the classifier with the worst accuracy from E.
      Add the new created classifier C_new into E.
    end if
    for each classifier in the ensemble C do
      for each instance x in chunk do
        Set m according to Poisson(1)
        Update m times each classifier member with x.
      end for
    end for
  end if
end while
```

observation, we construct a small heterogeneous ensemble rather than a big homogeneous ensemble with a large number of classifiers of the same type, which will compromise speed.

As mentioned, we aim to select online classifiers so that the ensemble can properly adapt to different types of concept drifts. Here, CVFDT [12] and Online Naive Bayes (OnlineNB) are chosen as the basic classifier types, but the framework can work with any classification algorithm. The OnlineNB is an online version of the Naive Bayes classier. When a training instance (x_t, y_t) comes, OnlineNB updates the corresponding prior and likelihood probabilities, $P(y_t)$ & $P(F_i = f_i^t | y_t)$. To classify a testing instance, it applies Bayes' theorem to select the class having the maximum posterior probability as follows:

$$OnlineNB(x_t) = argmax_{y_j} P(y_j) \prod_{i=1}^{n} P(F_i = f_i^t | y_j) \tag{5}$$

Details of the heterogeneous ensemble's learning process are given in Algorithm 1. Given a data stream X and a predefined set $\mathcal{M}$ of different classifier

Algorithm 2. Real Time Classification

Input: A new testing unlabeled instance x_t.
An ensemble $\mathcal{E}$ of N classifiers, denoted as $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_N$. Every classifier $\mathcal{C}_i$ is associated with a feature subset $\varphi_\mathbf{i}$.
Given a testing instance, every classifier $\mathcal{C}_i$ outputs a probability distribution vector $\rho_i = [\rho_{i1}, \rho_{i2}, \ldots, \rho_{ic}], (\sum_{j=1}^{c} \rho_{ij} = 1,\ i \in \{1, 2, \ldots, N\})$.
Output: The ensemble 's probability distribution vector for x_t.

1: **for** every classifier $\mathcal{C}_i$ in the ensemble $\mathcal{E}$ **do**
2: Project the arriving testing instance$\mathbf{x}_t$ onto a low dimensional feature space $\varphi_\mathbf{i}$ and get $\widehat{\mathbf{x}}_t$.
3: Compute the probability distribution vector $\rho_i = \mathcal{C}_i(\widehat{\mathbf{x}}_t)$.
4: Get the aggregated error of $\mathcal{C}_i$, err_i, and calculate the weight following the Equation 8, $w_i = 1/(err_i + \alpha)$.
5: **end for**
6: Aggregate all probability distribution vectors as follows:

$$\mathcal{E}(\rho) = \sum_{i=1}^{z} w_i * \rho_i$$

7: Normalize and return vector $\mathcal{E}(\rho)$.

types, we initialize the ensemble with k classifiers of each type in $\mathcal{M}$. Next, data streams are processed in a sliding window mode and data instances are grouped into predefined-size chunks. When a new chunk arrives, we apply a feature selection technique to find the most discriminative feature subset. If the subset is different from the previous one, there is a feature drift. We would then need to construct a new classifier with the selected feature subset. We add it to the ensemble, and remove the worst classifier if the ensemble is full; the new classifier will be of the same type as the best classifier member which has the smallest aggregated error (lines 7-12). Finally, we employ online bagging [15] for updating classifier members to reduce the variance of the ensemble (lines 13-18).

Ensemble Classification. Based on the research work of Tumer *et. al* [20] and Fumera *et. al* [10], the estimate error of the ensemble are:

$$E_{add}^{ens} = \sum_{k=1}^{N} w_k^2 E_{add}^k + \sum_{k=1}^{N} \sum_{l \neq k} w_l w_k \left[\beta^k \beta^l + \rho^{kl} (\sigma_i^k \sigma_i^l + \sigma_j^k \sigma_j^l)/s^2 \right], \tag{6}$$

where β^k and σ^k are the bias and the standard deviation of the estimate error ε^k of classifier C_k , ρ^{kl} is the correlation coefficient of the errors ε^k and ε^l.

We assume that classifier members are unbiased and uncorrelated (i.e. $\beta^k = 0, \rho^{kl} = 0, k \neq l$). Then, we have $E_{add}^{ens} = \sum_{k=1}^{N} w_k^2 E_{add}^k, \sum_{k=1}^{N} w_k = 1$. To minimize the added error of the ensemble, the weights of classifier members are set as follows:

$$w_k = (E_{add}^k)^{-1} \left[\sum_{m=1}^{N} (E_{add}^m)^{-1} \right] \tag{7}$$

When constructing the ensemble classifier, we estimate the added error of every classifier member, E^k_{add}, which is its accumulated error from its creation time to the current time. Moreover, to alleviate the extreme case of $E^m_{add} \approx 0$, we modify the Equation 7 as follows:

$$w_k = (E^k_{add} + \alpha)^{-1} \left[\sum_{m=1}^{N} (E^m_{add} + \alpha)^{-1} \right], \tag{8}$$

where α is a padding value which is empirically set to 0.001.

Details of the ensemble classification process are shown in Algorithm 2. Given the ensemble $\mathcal{E}$ where each member can classify a testing instance and output the result as a probability distribution vector, we use a weighting combination scheme to get the result. First, for each classifier $\mathcal{C}_i$ we project the testing instance x_t onto the subspace $\varphi_\mathbf{i}$. Then, the classifier $\mathcal{C}_i$ processes the projected instance and outputs a probability distribution vector. We attain the aggregated accuracy of $\mathcal{C}_i$ from its creation time, and calculate its optimal weight according to Equation 8 to minimize the expected added error of the ensemble. Finally, the ensemble's result is set as the normalized sum of the weighted classification results of the members (lines 6-7).

4 Experiments and Analysis

4.1 Experimental Setup

For our experiments, we use three synthetic datasets and three real life datasets. The three synthetic datasets, SEA generator (SEA), Rotating Hyperplane (HYP), and LED dataset (LED), are generated from the MOA framework [1]. Concept drifts are generated by moving 10 centroids at a speed of 0.001 per instance in the RBF dataset, and changing 10 attributes at a speed of 0.001 per instance in the HYP dataset. The three real life datasets are: network intrusion (KDD'99[1]), hand-written digit recognition (MNIST[2]), and protein crystallography diffraction (CRYST[3]) datasets. Table 1 shows the characteristics of the six datasets.

We compare our algorithm HEFT-Stream with other prominent ensemble methods: AWE [21] and OnlineBagging [15]. The experiments were conducted on a Windows PC with a Pentium D 3GHz Intel processor and 2GB memory. To enable more meaningful comparisons, we try to use the same parameter values for all the algorithms. The number of classifier members is set to 10 for all the ensemble algorithms, and the chunk size is set to 1000. To simulate the data stream environment, we process all experiments in a practical approach, called *Interleaved-Chunk*. In this approach, data instances are read to form a data chunk. Each new data chunk is first used to test the existing model. Then it is used to update the model and it is finally discarded to save memory.

[1] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[2] http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets
[3] http://ajbcentral.com/CrySis/dataset.html

Table 1. Characteristics of datasets used for evaluation

Name	#Instances	#Attributes	#Classes	Noise	#Selected Features	**Ratio of FS**
SEA	100,000	3	2	10%	2.25	**75%**
HYP	100,000	10	2	5%	6.71	**67.13%**
LED	100,000	24	3	10%	15.84	**65.98%**
KDD'99	494,022	34	5	N/A	2.33	**6.87%**
MNIST	60,000	780	10	N/A	30.77	**3.95%**
CRYST	5,500	1341	2	N/A	7.49	**0.56%**

4.2 Experimental Results

As AWE and OnlineBagging are homogeneous ensembles and can only work with one classifier type, we set different classifier types for these ensembles accordingly. For AWE, we set classifier members as Naive Bayes and C4.5, which are recommended by the authors [21], and denoted as *AWE(NB)* and *AWE(C4.5)*. For OnlineBagging, we set its classifier members as OnlineNB and CVFDT, and denoted as *Bagging(OnlineNB)* and *Bagging(CVFDT)* respectively. We conduct the experiments ten times for each dataset and summarize their average accuracy and running times in Table 2. Readers may visit our *website*[4] for algorithms' implementation, more experimental results, and detailed theoretical proofs.

We observe that the AWE ensemble has the worst accuracy and the longest running time. This is because the AWE ensemble uses traditional classifiers as its members and trains them once; when there are concept drifts, these members become outdated and accuracy is degraded. Moreover, the AWE ensemble always trains a new classifier for every upcoming chunk, and this increases processing time. The OnlineBagging has better performance but it largely depends on the classifier type. For example, Bagging(OnlineNB) is more accurate and faster than Bagging(CVFDT) for the LED dataset but Bagging(OnlineNB) become less precise and slower than Bagging(CVFDT) for the KDD'99 dataset. It is also noteworthy that both AWE and OnlineBagging do not work well with high dimensional datasets, such as MNIST and CRYST.

Our approach, HEFT-Stream, addresses the above problems and achieves better performance than WCE and OnlineBagging. It achieves the best accuracy values and the lowest running time for most datasets. HEFT-Stream continuously updates classifier members with gradual drifts, and only trains new classifiers whenever there are feature drifts or sudden drifts. This property not only enables HEFT-Stream to adapt to different types of concept drifts but also conserve computational resources. Furthermore, HEFT-Stream can dynamically change the ratio among classifier types to adapt to different types of datasets; when a particular classifier type works well with a certain dataset, its ratio in the ensemble will be increased. Finally, with integrated feature selection capability, HEFT-Stream only works with the most informative feature subsets which improves accuracy and reduces processing time. The last column of the Table 1

[4] http://www3.ntu.edu.sg/home2008/nguy0105/heftstream.html

Table 2. Comparisons of AWE(NB), AWE(C4.5), Bagging(OnlineNB), Bagging(CVFDT), and HEFT-Stream. Time is measured in seconds. For each dataset, the highest accuracy value is **boldfaced**, and the lowest running time is underlined.

Dataset	AWE(NB)		AWE(C4.5)		Bagging(OnlineNB)		Bagging(CVFDT)		HEFT-Stream	
	Acc	Time	Acc	Time	Acc	Time	Acc	Time	Acc	Time
SEA	88.08	6.61	88.12	26.08	87.91	2.9	89.12	21.47	**89.28**	22.65
HYP	87.94	19.42	72.72	27.40	86.92	8.07	88.90	40.41	**89.18**	12.42
LED	73.91	74.33	72.13	40.34	73.93	26.21	73.79	83.00	**74.07**	28.02
KDD'99	95.09	280.33	94.68	281.33	92.95	230.3	**97.75**	209.6	96.37	142.0
MNIST	9.87	2054.0	78.49	1246.7	9.87	1286.00	21.13	1456.00	**79.36**	439.00
CRYST	53.70	40.38	83.30	147.00	54.28	57.63	76.18	101.33	**83.52**	37.10
Average	68.10	412.51	81.57	294.80	67.64	268.53	74.48	318.65	**85.30**	113.53

shows the ratios of the selected features to the full feature sets for all datasets. We realize that feature selection techniques are very useful for high dimensional datasets. For example, the percentages of the selected features are 3.95% for the MNIST dataset, and only 0.56% for the CRYST dataset.

5 Conclusions

In this paper, we proposed a general framework to integrate feature selection and heterogeneous ensemble learning for stream data classification. Feature selection helps to extract the most informative feature subset which accelerates the learning process and increases accuracy. We first apply feature selection techniques on the data streams in a sliding window manner and monitor the feature subset sequence to detect feature drifts which represent sudden concept drifts. The heterogeneous ensemble is constructed from well-chosen online classifiers. The ratios of classifier types are dynamically adjusted to increase the ensemble 's diversity, and allows the ensemble to work well with many kinds of datasets. Moreover, the ensemble adapts to the severity of concept drifts; we update the online classifier members for gradual drifts, and replace an outdated member by a new one for sudden drifts. We have conducted extensive experiments to show that our ensemble outperforms state-of-the-art ensemble learning algorithms for data streams.

In our future work, we will investigate more intelligent methods to adjust the ratios of classifier types as well as the ensemble size. We will continue to examine the relationship between concept and feature drifts and develop a metric to quantify concept drifts and use it to further adapt ensembles to achieve better accuracy.

References

1. Bifet, A., Holmes, G., Kirkby, R.: Moa: Massive online analysis. The Journal of Machine Learning Research 11, 1601–1604 (2010)
2. Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., Gavald, R.: New ensemble methods for evolving data streams. In: 15th ACM SIGKDD, pp. 139–148. ACM (2009)

3. Breiman, L.: Bagging predictors. The Journal of Machine Learning Research 24(2), 123–140 (1996)
4. Breiman, L.: Random forests. The Journal of Machine Learning Research 45(1), 5–32 (2001)
5. Domingos, P., Hulten, G.: Mining high-speed data streams. In: The Sixth ACM SIGKDD, pp. 71–80. ACM (2000)
6. Domingos, P., Pazzani, M.: On the optimality of the simple bayesian classifier under zero-one loss. The Journal of Machine Learning Research 29(2-3), 103–130 (1997)
7. Eibl, G., Pfeiffer, K.-P.: Multiclass boosting for weak classifiers. The Journal of Machine Learning Research 6, 189–210 (2005)
8. Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In: The 13th ICML, pp. 148–156 (1996)
9. Friedman, J.H.: Stochastic gradient boosting. Computational Statistics & Data Analysis 38(4), 367–378 (2002)
10. Fumera, G., Roli, F.: A theoretical and experimental analysis of linear combiners for multiple classifier systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(6), 942–956 (2005)
11. Hsu, K.-W., Srivastava, J.: Diversity in Combinations of Heterogeneous Classifiers. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 923–932. Springer, Heidelberg (2009)
12. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: ACM SIGKDD, pp. 97–106. ACM (2001)
13. Lei, Y., Huan, L.: Feature selection for high-dimensional data: A fast correlation-based filter solution. In: The 20th ICML, pp. 856–863 (2003)
14. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering 17(4), 491–502 (2005)
15. Oza, N.C.: Online bagging and boosting. In: 2005 IEEE International Conference on Systems, Man and Cybernetics, vol. 3, pp. 2340–2345. IEEE (2005)
16. Sattar, H., Ying, Y., Zahra, M., Mohammadreza, K.: Adapted one-vs-all decision trees for data stream classification. IEEE Transactions on Knowledge and Data Engineering 21, 624–637 (2009)
17. Shen, C., Li, H.: On the dual formulation of boosting algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(12), 2216–2231 (2010)
18. Street, W.N., Kim, Y.: A streaming ensemble algorithm (sea) for large-scale classification. In: The 7th ACM SIGKDD, pp. 377–382. ACM (2001)
19. Tin Kam, H.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)
20. Tumer, K., Ghosh, J.: Linear and order statistics combiners for pattern classification. Springer (1999)
21. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: ACM SIGKDD, pp. 226–235. ACM (2003)
22. Woods, K., Philip Kegelmeyer, J.W., Bowyer, K.: Combination of multiple classifiers using local accuracy estimates. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(4), 405–410 (1997)
23. Zhenyu, L., Xindong, W., Bongard, J.: Active learning with adaptive heterogeneous ensembles. In: The 9th IEEE ICDM, pp. 327–336 (2009)

OMC-IDS: At the Cross-Roads of OLAP Mining and Intrusion Detection

Hanen Brahmi[1], Imen Brahmi[1], and Sadok Ben Yahia[1,2]

[1] LIPAH, Computer Science Department, Faculty of Sciences of Tunis, Tunis, Tunisia
[2] Institut TELECOM, TELECOM SudParis, UMR 5157 CNRS SAMOVAR, France
{hanenbrahmi,imen.brahmi}@gmail.com
sadok.benyahia@fst.rnu.tn

Abstract. Due to the growing threat of network attacks, the efficient detection as well as the network abuse assessment are of paramount importance. In this respect, the Intrusion Detection Systems (IDS) are intended to protect information systems against intrusions. However, IDS are plugged with several problems that slow down their development, such as low detection accuracy and high false alarm rate. In this paper, we introduce a new IDS, called OMC-IDS, which integrates data mining techniques and On Line Analytical Processing (OLAP) tools. The association of the two fields can be a powerful solution to deal with the defects of IDS. Our experiment results show the effectiveness of our approach in comparison with those fitting in the same trend.

Keywords: Intrusion detection system, Data warehouse, OLAP, Audit data cube, Association rules, Classification.

1 Introduction

As far as interconnections among computer systems grow rapidly, network security is becoming a major challenge. An *Intrusion Detection System* (IDS) has been of use to monitor the network traffic thereby detect whether a system is being targeted by network attacks [14]. Even that IDSs have become a standard component in security infrastructures, they still have a number of significant drawbacks [14]. Indeed, the volume of the audit data which an IDS has to monitor is huge and grows rapidly. In addition, they flag out lower accuracy and higher false alarm rates. Moreover, current IDS do not provide support for historical data analysis and data summarization [13]. Supporting a historical network database in conjunction with an IDS raises two important technical challenges [8]: (*i*) since network traffic monitors generate data continuously and at high-rate, the database needs to support a high data insertion rate [8]; (*ii*) to facilitate the security analysis, the database must quickly answer historical queries [8,13].

Recently, *Data Warehouses* (DW) and *On Line Analytical Processing* (OLAP) technologies have gained a widespread acceptance as a support for decision making [7]. In a DW architecture, data are manipulated through OLAP tools which

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 13–24, 2012.

offer visualization and navigation mechanisms of multidimensional data views, commonly called *data cubes* [7]. Along with the increasing complexity of networks, protecting a system against new and complex attacks, while keeping an automatic and adaptive framework, is a thriving issue. One answer to the problem could rely on the association of OLAP and data mining to allow elaborated analysis tasks exceeding the simple exploration of the traffic data. DW and OLAP techniques can help the security officer in detecting attacks, monitoring current activities on the network, historical data analysis about critical attacks in the past and generating reports on trend analysis [13]. While, data mining is known for its ability to discover knowledge from audit data [14].

In this paper, we investigate another way of tackling the aforementioned problems. Thus, we introduce a new IDS based on a DW perspective to enhance the accuracy of detection as well as to minimize the false alarm rates. To that end, our proposed system integrates the OLAP and data mining techniques to improve the performance and usability of an IDS. Firstly, we model the network traffic data as a multidimensional structure, called *audit data cube*. Secondly, we introduce a novel algorithm that provides a concise representation of multidimensional association rules mined from the audit data cube. Finally, a classifier is used to decide whether a new connection record is an attack or not using the set of multidimensional detection rules. Through extensive carried out experiments on the standard intrusion detection DARPA dataset, we show the effectiveness of our proposal on the IDS performance aspects related to the false alarms as well as the detection rates.

The remaining of the paper is organized as follows. Section 2 sheds light on some representative related work applying the data mining techniques into the IDS. We introduce our new IDS based on the OLAP and data mining techniques in Section 3. We also relate the encouraging results of the carried out experiments in Section 4. Finally, Section 5 concludes and points out avenues of future work.

2 Scrutiny of the Related Work

Before data mining techniques are introduced into the intrusion detection field, the latter was heavily dependent on a manually maintained knowledge basis to reflect the ever-changing situations. However, this traditional way is difficult and expensive [14]. Otherwise, within data mining techniques, the rules (or signatures) of normal and abnormal activities can be created automatically. It is also possible to detect new types of attacks through an incremental learning process. Additionally, data mining techniques provide the means to easily perform data summarization and visualization, that would be of great help to the security analyst in identifying areas of concern [14]. In the following, we survey the most prominent approaches dedicated to apply data mining techniques within the intrusion detection field.

- The **MADAM-ID system** [10] is considered as the first research work that shows how data mining techniques can be used to construct IDS in a more systematic and automated manner. Firstly, all network traffic is abstracted to connection records. The latter are classified into “normal” and “intrusion”.

- The **ADAM system** [2] is one of the best-known approaches that use association rules mining and classification algorithms to detect intrusions. The main moan that can be addressed to ADAM stands in its high dependency on training data for normal activities. However, the attack-free training data is difficult to afford, since there is no guarantee that we can prevent all attacks in real world networks.
- The **MINDS system** [6] allows the development of scalable data mining algorithms and tools for detecting attacks and threats against computer systems. In fact, the system clusters audit data using a density-based local outliers algorithm to detect intrusions. In addition, it applies an association pattern analysis to summarize the network connections that are highly ranked as anomalous by the algorithm.

On the one hand, although the data mining techniques could provide beneficial characteristics to IDS, there is a compelling need to develop methods and tools that can help in historical data analysis. On the other hand, within a typical network environment, many different audit streams, collected from multiple cyber sensors, are shown to be useful for detecting intrusions. Such data includes: (*i*) raw network traffic data; (*ii*) netflow data; (*iii*) system calls; and so on. Consequently, it is important to have an architecture that can integrate these heterogenous data sources into a unified framework. The research works of [13] focus on the OLAP techniques to represent network traffic data and relate it to the corresponding IDS alerts. In contrast, we propose to couple OLAP and data mining techniques for intrusion detection. The main idea behind our approach is to take advantage from OLAP as well as data mining techniques and to integrate them to the same analysis framework in order to improve the performance of an IDS. In this paper, we introduce a new IDS, called OMC-IDS (Olap *Mining and Classification-based IDS*), which affords a support for historical data analysis and data summarization as well as the capacity to handle any kind of data for intrusion detection.

3 OMC-IDS: Intrusion Detection Based on Olap Mining and Classification

The OMC-IDS enriches the OLAP techniques with data mining facilities to benefit from their cross capabilities they offer. Indeed, the audit data collected from different heterogenous resources goes through four stages. Firstly, the data is filtered to remove irrelevant information and a relational database is created containing the meaningful remaining data. This database facilities information extraction and data summarization based on individual attributes such as day, source, destination, etc. Secondly, an audit data cube is constructed using the available dimensions. Thirdly, the OMC-IDS system integrates OLAP technology and association rule mining in order to extract interesting information under different perspectives and levels of granularity. Finally, OMC-IDS uses a classifier to classify each connection record either as one of the attack types or normal.

In the following, we focus on the study of the three last steps of the OMC-IDS system.

3.1 Audit Data Cube: Construction and Manipulation

The data feeding a data warehouse and OLAP systems is usually organized into multidimensional data views commonly called *data cubes*. The latter contain fact tables related to several dimension tables. A fact table represents the focus of analysis and typically includes attributes called *measures*. These are usually numerical values that facilitate a quantitative evaluation of various aspects of interest. Dimensions include attributes that form hierarchies. As long as a hierarchy is traversed from finer to coarser levels, measures are aggregated. Hierarchies can be included in a flat table forming the so-called *STAR schema* [7].

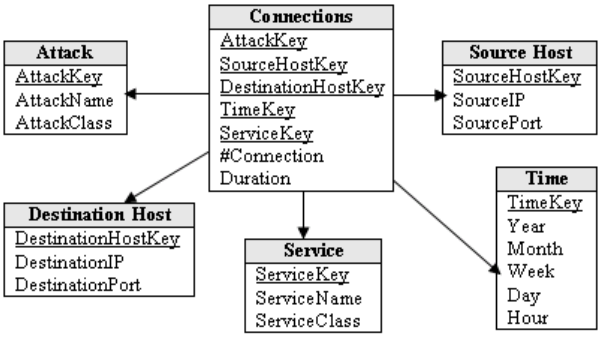


Fig. 1. A STAR schema for the IDS data warehouse

We propose to model the audit data as a multidimensional structure based on the STAR schema shown in Figure 1. The fact table "Connections" contains the attribute "#Connection" that measures the number of connections. The dimension "Time" includes information of date and time when the network packet was captured. The dimension "Service" contains the name and the class of service (or protocol) that was attacked. "Source Host" describes the source of IP addresses and port number. Likewise, the dimension "Destination Host" describes the destination of IP address and port. Similarly, the dimension "Attack" contains both the name of the attack and its type. Furthermore, hierarchies would give an extra edge for analysis purpose, since they allow decision-making users to see quantified data at different levels of abstraction. Therefore, security analysts must deal with hierarchies to exploit OLAP systems to their fullest capabilities. To do so, we define a concept hierarchy for each dimension in the audit data cube. For example, "Hour $\rightarrow$ Day $\rightarrow$ Week $\rightarrow$ Month $\rightarrow$ Year" is the hierarchy on the "Time" dimension. The dimension "Attack" can be organized into the hierarchy "Name $\rightarrow$ Class", *e.g.*, "*Smurf* $\rightarrow$ *DoS*". In addition, the hierarchies can be pre-defined or generated by partitioning the dimension into ranges. For instance, the dimension "Duration" could be partitioned into categories as "Low", "Medium" and "High".

Using the STAR schema described in Figure 1, a corresponding audit data cube would be a six dimensional structure in which a cell contains aggregates of the operations measures. For instance, a cell could correspond to short duration attacks over the FTP service in the period 1 pm to 2 pm on Oct. 20th 2011. The audit data cube can be constructed by using the SQL aggregation functions (*e.g.*, COUNT, SUM, MIN, MAX). For example, the COUNT value refers to the number of connections. The audit data can be manipulated with great flexibility and viewed from different perspectives by the use of data cubes. Indeed, OLAP operations (*e.g.*, ROLL-UP, DRILL-DOWN, SLICE and DICE) offer analytical modeling capabilities that can be applied on the audit data. The ROLL-UP operation allows the going from specific to general by climbing up the aggregation hierarchy. Otherwise, going from generalized data to more specific by stepping down the aggregation hierarchy is called DRILL-DOWN. The SLICE and DICE operations reduce the dimensionality of data by projecting the data on a subset of dimensions for selected values of other dimensions.

3.2 Multidimensional Association Rule Mining

The association rule extraction is a technique of data mining to discover interesting correlation relationships among data. In fact, the formalization of the association rule mining problem was initially introduced by Agrawal et *al.* [1]. Given a set of records, the objective of mining association rules is to extract all rules of the form X ⇒ Y that satisfy a user-specified minimum support and minimum confidence thresholds, *i.e.*, *minSup*[1] and *minConf*[2]. X is the antecedent of the rule and Y is its consequent.

In the recent years, the problem of mining association rules from data cubes is knowing an increasing interest. The association rule mining can make OLAP more useful and easier to apply in the overall scheme of decision support systems. Further, OLAP is closely interlinked with association rules and shares with them the goal of finding patterns in the data. Indeed, data cube structures make good use of aggregated data, at the desired granularity levels, in the computation of the support and the confidence [3].

The multidimensional association rules is shown to be useful in increasing the detection accuracy and decreasing the false positives rate [12]. Consequently, the IDS performances can be greatly improved whenever the association rules are mined from the audit data cube. However, the number of the mined rules can be quite large, which affects the speed of IDS and hampers its whole performance [6,12]. Some of these rules are redundant since they contain patterns that correspond to the subsets of other patterns.

Example 1. Let R and R_1 tow multidimensional association rules. R: {Src_Port = 21 ∧ Dst_IP = 192.63.11.11 ∧ service = telnet ∧ Duration = Long} ⇒ {Attack = *Smurf*} and R_1: {Src_Port = 21 ∧ service = telnet} ⇒ {Attack = *Smurf*}. R and R_1 share similar features, *i.e.*, the patterns "Src_Port = 21" and "service =

[1] *minSup* refers to the minimum support threshold pre-defined by the user.
[2] *minConf* refers to the minimum confidence threshold pre-defined by the user.

telnet". If the respective supports of these two patterns are equal, then the rule R_1 is redundant *w.r.t* R.

To effectively mine the non-redundant multidimensional association rules from the audit data cube, we use the concept of closure [11] defined as follows:

Definition 1. *A pattern X is a closed pattern if there exists no pattern X' such that: (i) X' is a proper superset of X; and (ii) every connection record in a network traffic containing X also contains X'. The closure γ of a pattern X is the maximal superset of X having the same support value as that of X.*

In this respect, we introduce the AMAR (*Audit Multidimensional Association Rules mining*) algorithm intended to mine a concise representation of multidimensional association rules from an audit data cube $\mathcal{AC}$. The pseudo-code is shown by Algorithm 1.

Algorithm 1. The AMAR algorithm.

Input: $\mathcal{AC}$, $\mathcal{D}$, $\mathcal{H}_{\mathcal{D}}$, *minSup*, *minConf*.
Output: Set of multidimensional non-redundant association rules, *i.e.*, $\mathcal{X} \Rightarrow \mathcal{Y}$, with corresponding Supp and Conf.

```
1  Begin
2    C_1 := {1-candidate};
3    k := 1; /* |1-candidate| is the cardinality of attributes corresponding to D and
     H_D.*/
4    While C_k ≠ ∅ and k ≤ |1-candidate| do
5      CC_k := ∅;
6      FC_k := ∅;
7      Foreach candidate pattern A ∈ C_k do
8        CC_k := CC_k ∪ γ(A);
9      Foreach candidate closed pattern A ∈ CC_k do
10       Supp := ComputeSupport(A);
11       If Supp ≥ minSup then
12         FC_k := FC_k ∪ A;
13     Foreach A ∈ FC_k do
14       Foreach B ≠ ∅ and B ⊂ A do
15         Conf := ComputeConfidence(A − B, B);
16         If Conf ≥ minConf then
17           X := A − B;
18           Y := B;
19           return (X ⇒ Y, Supp, Conf);
20     C_{k+1} := ∅;
21     Foreach A ∈ FC_k do
22       Foreach B ∈ FC_k that shares (k-1) items with A do
23         If All Z ⊂ {A ∪ B} of k items are inter-dimensional and closed
           frequent then
24           C_{k+1} := C_{k+1} ∪ {A ∪ B};
25     k := k + 1;
26 End
```

Usually the user is interested in specified subsets of attributes in order to extract interesting relationships among them. So, (s)he needs to exclude the set of irrelevant attributes from the examination. To that end, AMAR allows the user to guide the analysis process by: (*i*) defining the set of dimensions $\mathcal{D}$ to be analyzed; (*ii*) choosing the hierarchies levels $\mathcal{H}_{\mathcal{D}}$ associated to the analysis dimensions; and (*iii*) setting the *minSup* and the *minConf* thresholds. As

sketched by Algorithm 1, we proceed by a bottom-up level wise search for frequent closed k-patterns, where the level k is the number of items in the set. We denote by C_k the sets of k-patterns that are potentially closed, CC_k the sets of closed k-patterns that are potentially frequent and FC_k the sets of frequent closed k-patterns. During the **initialization step** (line 2), our algorithm captures the 1-candidates from the user defined analysis dimensions $\mathcal{D}$ over the audit data cube $\mathcal{AC}$. These 1-candidates correspond to the attributes of $\mathcal{D}$, where each one complies with the chosen hierarchies $\mathcal{H}_{\mathcal{D}}$.

Within the **first step**, AMAR applies the closure concept (*cf.* Definition 1). The **second step** (lines 9-12) of our algorithm derives the frequent closed patterns FC_k from the closed candidate patterns CC_k that have a support greater or equal to $minSup$. The **third step** (lines 13-19) allows the extraction of association rules with a confidence greater or equal to $minConf$. The computation of support and confidence are performed respectively by the ComputeSupport and ComputeConfidence functions. Both functions directly pick up required precomputed aggregates from the data cube via MDX (MultiDimensional eXpression) queries [3]. The **fourth step** (lines 20-24) uses the set of frequent closed k-patterns FC_k to derive a new set of $(k+1)$-candidates, denoted by C_{k+1}. One $(k+1)$-candidate is the union of two k-patterns $\mathcal{A}$ and $\mathcal{B}$ from FC_k that respects three conditions: (i) $\mathcal{A}$ and $\mathcal{B}$ must have k-1 common patterns; (ii) all non empty sub-patterns from $\mathcal{A} \cup \mathcal{B}$ must be instances of inter-dimensional[3] patterns in $\mathcal{D}$; and (iii) all non empty sub-patterns from $\mathcal{A} \cup \mathcal{B}$ must be frequent closed patterns.

Table 1. A snapshot of an audit data cube with four dimensions

Service	Src_Port	Dst_Port	Attack	#Con
Imap	63587	143	*Neptune*	44
Imap	6161	143	*Satan*	26
Pop3	6161	110	*Neptune*	15
Pop3	63587	143	*Satan*	20
Tcpmux	63587	1	*Neptune*	64

Table 2. Multidimensional association rule list

ID	Rules	Sup	Conf
R_1	143 ⇒ *Satan*	0.3	0.5
R_2	143 ∧ 63587 ⇒ Imap ∧ *Neptune*	0.3	0.7
R_3	*Satan* ⇒ Imap ∧ 143 ∧ 6161	0.2	0.6
R_4	63587 ∧ *Neptune* ⇒ Tcpmux ∧ 1	0.4	0.6
R_5	Pop3 ∧ 143 ⇒ 63587 ∧ *Satan*	0.1	1.0
R_6	Pop3 ∧ 63587 ⇒ 143 ∧ *Satan*	0.1	1.0

Example 2. Table 1 sketchs an example of an audit data cube with four dimensions. The last row measures the number of connections using the aggregation function Count. The set of closed patterns, with their corresponding supports, is as follows: {("Pop3": 0.2), ("143":0.5), ("63587":0.7), ("6161": 0.2), ("*Neptune*": 0.7), ("Pop3, 143, 63587, *Satan*": 0.1), ("Pop3, 110, 6161, *Neptune*": 0.08), ("63587, 143": 0.3), ("143, *Satan*": 0.2), ("Imap, 143": 0.4), ("63587, *Neptune*": 0.6), ("Imap, 143, 6161, *Satan*": 0.1), ("Imap, 143, 63587, *Neptune*": 0.2), ("Tcpmux, 1, 63587, *Neptune*": 0.3)}. We extract the set of multidimensional association rules using the AMAR algorithm. Throughout our example, we set the $minSup$ to 10% and the $minConf$ to 50%. The algorithm generated 40 rules. Some of the extracted rules are illustrated in table 2.

[3] An inter-dimensional pattern is composed of items coming from different dimensions.

4 Classification

Intrusion detection can be considered as a classification problem where each connection is identified either as one of the attack types or normal based on some existing data [13]. Some of the association rules extracted by the AMAR algorithm are not useful since they do not imply an intrusion type in their consequent part. Therefore, we select the set of rules whose consequents include an intrusion label. For instance, according to the set of rules illustrated by Table 2, the rules R_3 and R_4 are excluded to retain only the rules R_1, R_2, R_5 and R_6. Then, we apply a decomposition axiom introduced in [4] (*cf.* Definition 2) to obtain new rules of the form "feature$_1$ $\wedge$ feature$_2$ $\wedge \ldots \wedge$ feature$_n$ $\Rightarrow$ intrusion". Even though, the obtained rules are redundant, their generation is mandatory to guarantee a maximal cover of the necessary rules.

Definition 2. *Given an association rule R, a decomposition axiom is defined as follows: If $R : X \Rightarrow Y$ then $R_1 : X \Rightarrow Z$ is a derivable valid rule, $\forall Z \subset Y$.*

Example 3. Let us consider the rule R_2: {Dst_Port = 143 $\wedge$ Src_Port = 63587} $\Rightarrow$ {Service = Imap $\wedge$ Attack = *Neptune*}. Using the decomposition axiom, R_2 is transformed in R'_2: {Dst_Port = 143 $\wedge$ Src_Port = 63587} $\Rightarrow$ {Service = Imap} and R''_2: {Dst_Port = 143 $\wedge$ Src_Port = 63587} $\Rightarrow$ {Attack = *Neptune*}. We retain the rule R''_2, since it includes an intrusion label in its consequent part.

Whenever the rules imply the same intrusion, we retain the rule which poses less constraints and can match more audit records.

Example 4. Let us consider two rules R: {Service = frag $\wedge$ Src_IP = 209.30.71.165 $\wedge$ Src_port = 110 $\wedge$ Dst_port = 32} $\Rightarrow$ {Attack = *Pod*} and R_1: {Service = frag, Dst_port = 32} $\Rightarrow$ {Attack = Pod}. Both rules R and R_1 imply the same intrusion label (*i.e.*, "Attack = *Pod*"). R_1 is considered to be more interesting than R, since it is needless to satisfy the features "Src_IP = 209.30.71.165" and "Src_port = 110" to highlight the attack "*Pod*". Hence, R_1 implies less constraints and can match more connection records than R.

Once the detection rules are generated, the OMC-IDS system applies a classifier [5] to classify the new connection records. Indeed, while having a new connection record C$_{New}$, the detection of an intrusion consists in traversing the detection rules from up to down in the classifier. The first reached rule, whose antecedent's part corresponds (*i.e.*, included or equal) to the features of C$_{New}$, will be of use. Thus, C$_{New}$ will obtain the conclusion of the rule which indicates an attack.

Example 5. Let us consider a new connection record C$_{New}$: "service = frag, Src_IP = 209.30.71.165, Dst_port = 32". If we have in the classifier just the rule R (*c.f.* Example 4), we cannot classify C$_{New}$ since the attribute "Src_port = 110" does not permit the matching. However, the rule R_1 (*c.f.* Example 4), which has a smaller antecedent than R, can classify C$_{New}$.

The latter example shows that the AMAR algorithm provides the relevant set of detection rules of need for the classification step of OMC-IDS. In fact, the use of such set of rules is of benefit for classifying new connection records.

5 Experimental Results

To evaluate the effectiveness and efficiency of our proposed system OMC-IDS, we carried out extensive experiments on a PC equipped with a 3 GHz Pentium IV and 2 Go of main memory running under Linux Fedora Core 6. Indeed, we compare our approach with the pioneering approaches falling within the intrusion detection-based classification trend, namely, ADAM [2] and C4.5[4] [9]. During the carried out experiments, we use the DARPA1998[5] dataset. The latter consists of training data and test data. The training data are generated in the first seven weeks and testing data are derived in the rest two weeks. The attacks consisting of a total of 33 different attack types are divided into four different attack categories, namely *DoS*, *R2L*, *U2R* and *Probing*. To build the audit data cube, we use the seven weeks' training data. To that end, we adopt the STAR schema showed in Figure 1. The audit data cube construction is done using the Analysis Services of SQL Server 2008.

Through these experiments, we put the focus on the assessment of the IDS performances in terms of detection and false alarms rates.

1. The *Detection Rate* (DR) is the number of correctly detected intrusions;
2. The *False alarms Rate* (FR) is the number of normal instances that were incorrectly considered as attacks.

Table 3. The DR (%) of OMC-IDS with respect to the dimension's variation

Dimensions	*DoS*	*Probe*	*U2R*	*R2L*
2-D	96.8	86.4	66.6	74.9
3-D	97.9	83.2	67.8	76.7
4-D	98.2	91.1	69.8	79.5
5-D	98.5	**95.3**	71.5	81.3
6-D	**99.5**	95.2	**74.9**	**86.6**

Table 3 shows the DR of OMC-IDS with respect to the dimension's variation for the four attack categories. The dimensions variation was established using the AMAR algorithm. From the results, we can remark that the dataset with six dimensions gives the best performances to detect the *DoS* class with 99.5% DR whereas the dataset with five dimensions gives the worst DR with 98.5%. Moreover, the dataset with five dimensions generates the best performance to detect the *Probe* class with 95.3% DR. The 6-D dataset gives the best performance to detect the *U2R* class with 74.9% DR and 5-D generates the worst performance with only 71.5%. Finally, the DR of *R2L* class on the 6-D dataset is the highest one, *i.e.*, 86.6% while on the 5-D we have the worst performance with only 81.3% DR. As consequence, OMC-IDS allows the detection of the attacks with best DR as far as the number of dimensions is the highest one, *i.e.*, six dimensions. Even

[4] Available in Weka: http://www.cs.waikato.ac.nz/ml/weka/

[5] Available at http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1998data.html

though, the DR decreases according to the decrease of the dimension number, it is still high.

The main challenge of IDS is to increase the value of the DR, while decreasing the value of FR. Figure 2 presents the DR and the FR, obtained respectively by, OMC-IDS, ADAM and C4.5-based systems. It can be seen that our approach drastically outperforms the other ones. In fact, Figure 2 (A) shows that OMC-IDS achieves a total DR above 99%, 97%, 86% and 74%, respectively corresponding to the detection of four attack categories (*i.e.*, *DoS*, *Probe*, *R2L* and *U2R*).

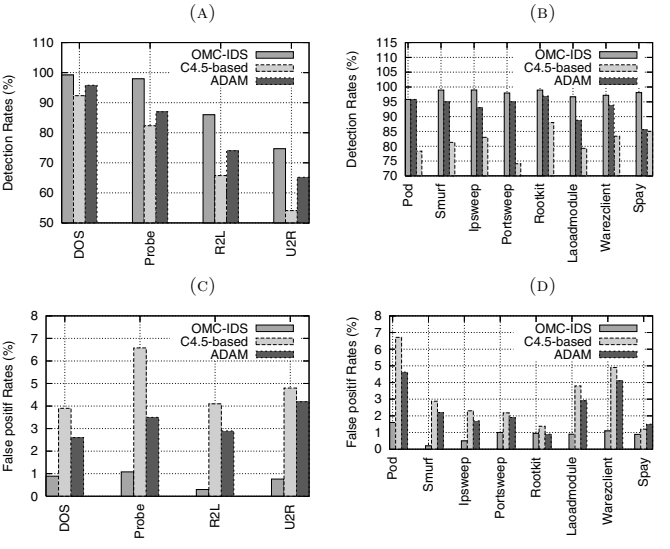


Fig. 2. Performance of OMC-IDS *vs.* ADAM and C4.5-based

Compared to ADAM, we remark that OMC-IDS provides a higher successful DR. Indeed, we achieved an average DR of 89% compared to 71%, over the four attack categories. On one hand, the high value of DR is explained by the use of pruning techniques that reduce the search space. In fact, the closed patterns have been shown to present the best compactness rates [11]. Thus, the mechanism adopted by the AMAR algorithm is more effective than that adopted by ADAM which is hampered by the ineffectiveness of the redundant association rules. On the other hand, the use of multidimensional association rules helps in improving the performance of detecting attacks. For example, let us consider the multidimensional association rule "{Src_Port = 21 $\wedge$ Dst_Port = 63 $\wedge$ Src_IP = 209.30.71.165 $\wedge$ Dst_IP = 180.66.11.11 $\Rightarrow$ Attack = *Satan*}". Obviously, the latter rule has higher accuracy than a single dimensional association rule "{Dst_IP = 180.66.11.11 $\Rightarrow$ Attack = *Satan*}". Consequently, we conclude that OMC-IDS is more efficient than ADAM due to the use of OLAP tools. In fact, the mining of multidimensional association rules from audit data cubes enhances the IDS process. Among the three tested systems, the C4.5-based IDS has the lowest

DR for the four attack classes. For instance, whenever OMC-IDS and ADAM have 74% and 65% DR for the *U2R* attacks, respectively, C4.5-based system has 54% DR. This is due to the stealthy nature of those attacks. Moreover, it is shown that C4.5 can classify more accurately on smaller datasets [9]. The results illustrated by Figure 2 (A) are confirmed by Figure 2 (B). The latter presents the DR of eight different attacks, including *Pod*, *Smurf*, *Ipsweep*, *Portsweep*, *Warezclient*, *Spay*, *Rootkit* and *Loadmodule*.

In addition, Figure 2 (C) shows that the FR ranges from 0.2% to 1%. The lowest FR is achieved for *DoS* attacks. The highest FR of *R2L* attacks generated by OMC-IDS is equal to 0.2%, which is a very low value compared to ADAM and C4.5-based systems. Precisely, according to Figure 2 (D), it is clear that the improvement of OMC-IDS with respect to ADAM is of 2%, 1.2%, 2% and 3%, respectively corresponding to the FR of the attacks *Smurf*, *Ipsweep*, *Loadmodule* and *Warezclient*.

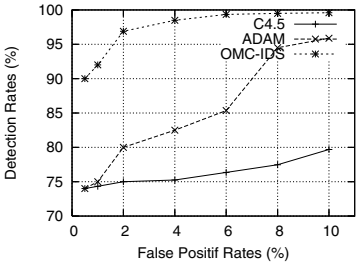


Fig. 3. The ROC curves of OMC-IDS *vs.* ADAM and C4.5-IDS

Within intrusion detection, the ROC (*Receiver Operating Characteristic*) curve is often used to assess the performance of IDSs. Figure 3 compares the ROC curve of OMC-IDS *vs.* those of ADAM and C4.5-based systems. It can be seen that the DR grows quickly to its peak value within a small increase of FR. In addition, the result ensures that our system can achieve the highest DR with the lowest FR. Thus, we conclude that OMC-IDS is more effective than ADAM and C4.5-based systems due to the use of the OLAP techniques that helped in improving the performance of detecting attacks.

6 Conclusion and Perspectives

On Line Analytical Processing (OLAP) provides tools to explore data cubes in order to extract interesting information. In this paper, we have shown the potential of coupling OLAP and data mining techniques in order to improve IDSs. To that end, we designed a new architecture, called OMC-IDS, to model network traffic using a multidimensional data structure based on the STAR schema. Carried out experiments showed that OMC-IDS outperforms the pioneering approaches, *i.e.*, ADAM and C4.5-based systems. Future work will

include exploring the alert correlations to expand the capabilities of our system. We can combine data from multiple sources to obtain a better analysis of the alert correlations.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: Proceedings of the ACM-SIGMOD International Conference on Management of Data, Washington, USA, pp. 207–216 (1993)
2. Barbara, D., Couto, J., Jajodia, S., Popyack, L., Wu, N.: ADAM: Detecting Intrusions by Data Mining. In: Proc. of the 2nd Annual IEEE SMC Information Assurance Workshop, West Point, NY, pp. 11–16 (2001)
3. Ben Messaoud, R., Rabaséda, S.L., Missaoui, R., Boussaid, O.: OLEMAR: An Online Environment for Mining Association Rules in Multidimensional Data, vol. 2, pp. 14–47 (2008)
4. Yahia, S.B., Nguifo, E.M.: Revisiting Generic Bases of Association Rules. In: Kambayashi, Y., Mohania, M., Wöß, W. (eds.) DaWaK 2004. LNCS, vol. 3181, pp. 58–67. Springer, Heidelberg (2004)
5. Brahmi, I., Ben Yahia, S., Slimai, Y.: IDS-GARC: Détection d'Intrusions Basée sur les Règles Associatives Génériques de Classification. In: Actes du 9ème Colloque Africain sur la Recherche en Informatique, Rabat, Maroc, pp. 667–674 (2008)
6. Chandola, V., Eilertson, E., Ertoz, L., Simon, G., Kumar, V.: Data Mining for Cyber Security. In: Singhal, A. (ed.) Data Warehousing and Data Mining Techniques for Computer Security, pp. 83–103. Springer (2006)
7. Chaudhuri, S., Dayal, U.: An Overview of Data Warehousing and OLAP Technology. SIGMOD Record 26(1), 65–74 (1997)
8. Geambasu, R., Bragin, T., Jung, J., Balazinska, M.: On-Demand View Materialization and Indexing for Network Forensic Analysis. In: Proceedings of the 3rd USENIX International Workshop on Networking Meets Databases, Cambridge, MA, pp. 4:1–4:7 (2007)
9. Gyanchandani, M., Yadav, R.N., Rana, J.L.: Intrusion Detection Using C4.5: Performance Enhancement by Classifier Combination. In: Proceedings of the International Conference on Advances in Computer Science, pp. 130–133 (2010)
10. Lee, W.: A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems. Phd thesis, Columbia University, New York, NY, USA (1999)
11. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Efficient Mining of Association Rules Using Closed Itemset Lattices. Journal of Information Systems 24(1), 25–46 (1999)
12. Ping-Ping, M., Qiu-Ping, Z.: Association Rules Applied to Intrusion Detection. Wuhan University Journal of Natural Sciences 7(4), 426–430 (2002)
13. Singhal, A.: Warehousing and Data Mining Techniques for Cyber Security. Advances in Information Security, vol. 31. Springer (2007)
14. Singhal, A., Jajodia, S.: Data Mining for Intrusion Detection. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 1171–1180. Springer (2010)

Towards Linear Time Overlapping Community Detection in Social Networks

Jierui Xie and Boleslaw K. Szymanski

Rensselaer Polytechnic Institute
Troy, New York 12180, USA
{xiej2,szymansk}@cs.rpi.edu

Abstract. Membership diversity is a characteristic aspect of social networks in which a person may belong to more than one social group. For this reason, discovering *overlapping* structures is necessary for realistic social analysis. In this paper, we present a fast algorithm[1], called SLPA, for overlapping community detection in large-scale networks. SLPA spreads labels according to dynamic interaction rules. It can be applied to both *unipartite* and *bipartite* networks. It is also able to uncover overlapping *nested hierarchy*. The time complexity of SLPA scales *linearly* with the number of edges in the network. Experiments in both synthetic and real-world networks show that SLPA has an excellent performance in identifying both *node* and *community* level overlapping structures.

1 Introduction

Community or modular structure is considered to be a significant property of real-world social networks. Thus, numerous techniques have been developed for effective community detection. However, most of the work has been done on *disjoint* community detection. It has been well understood that people in a real social network are naturally characterized by *multiple* community memberships. For example, a person usually has connections to several social groups like family, friends and colleges; a researcher may be active in several areas; in the Internet, a person can simultaneously subscribe to an arbitrary number of groups.

For this reason, overlapping community detection algorithms have been investigated. These algorithms aim to discover a *cover* [7], defined as a set of clusters in which each node belongs to at least one cluster. In this paper, we propose an efficient algorithm for detecting both individual overlapping nodes and overlapping communities using the underlying network structure alone.

2 Related Work

We review the state of the art and categorize existing algorithms into five classes that reflect how communities are identified.

Clique Percolation: CPM [11] is based on the assumption that a community consists of fully connected subgraphs and detects overlapping communities by

[1] Available at https://sites.google.com/site/communitydetectionslpa/

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 25–36, 2012.

searching for *adjacent* cliques. CPMw [4] extends CPM for weighted networks by introducing a subgraph intensity threshold.

Local Expansion: The iterative scan algorithm (IS) [2],[6] expands small cluster cores by adding or removing nodes until a local density function cannot be improved. The quality of seeds dictates the quality of discovered communities. LFM [7] expands a community from a random node. The size and quality of the detected communities depends significantly on the resolution parameter of the fitness function. EAGLE [14] and GCE [9] start with all maximal cliques in the network as initial communities. EAGLE uses the agglomerative framework to produce a dendrogram in $O(n^2 s)$ time, where n is the number of nodes, and s is the maximal number of join operations. In GCE communities that are similar within a distance ϵ are removed. The greedy expansion takes $O(mh)$ time, where m is the number of edges, and h is the number of cliques.

Fuzzy Clustering: Zhang [16] used the spectral method to embed the graph into low dimensionality Euclidean space. Nodes are then clustered by the fuzzy c-mean algorithm. Psorakis et al. [12] proposed a model based on Bayesian non-negative matrix factorization (NMF). These algorithms need to determine the number of communities K and the use of matrix multiplication makes them inefficient. For NMF, the complexity is $O(Kn^2)$.

Link Partitioning: Partitioning links instead of nodes to discover communities has been explored, where the node partition of a link graph leads to an edge partition of the original graph. In [1], single-linkage hierarchical clustering is used to build a link dendrogram. The time complexity is $O(nk_{max}^2)$, where k_{max} is the highest degree of the n nodes.

Dynamical Algorithms: Label propagation algorithms such as [13],[5],[15] use labels to uncover communities. In COPRA [5], each node updates its belonging coefficients by *averaging* the coefficients from all its neighbors in a synchronous fashion. The time complexity is $O(vm \log(vm/n))$ per iteration, where parameter v controls the maximum number of communities with which a node can associate, m and n are the number of edges and number of nodes respectively.

3 SLPA: Speaker-Listener Label Propagation Algorithm

Our algorithm is an extension of the Label Propagation Algorithm (LPA) [13]. In LPA, each node holds only a single label and iteratively updates it to its neighborhood majority label. Disjoint communities are discovered when the algorithm converges. Like [5], our algorithm accounts for *overlap* by allowing each node to possess multiple labels but it uses different dynamics with more general features.

SLPA mimics human pairwise communication behavior. In each communication step, one node is a speaker (information provider), and the other is a listener (information consumer). Unlike other algorithms, each node has a *memory* of the labels received in the past and takes its content into account to make the

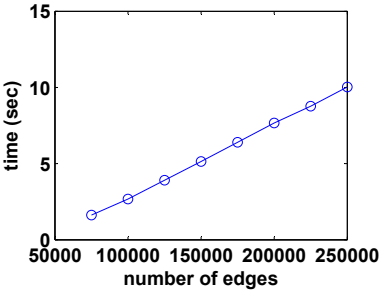


Fig. 1. The execution times of SLPA in synthetic networks with $n = 5000$ and average degree $\overline{k}$ varying from 10 to 80

current decisions. This allows SLPA to avoid producing a number of small size communities as opposed to other algorithms. In a nutshell, SLPA consists of the following three stages:

Algorithm 1. SLPA(T, r)

T: the user defined maximum iteration
r: post-processing threshold
1) First, the *memory* of each node is initialized with a unique label.
2) Then, the following steps are repeated until the maximum iteration T is reached:

   a. One node is selected as a listener.
   b. Each neighbor of the selected node randomly selects a label with probability proportional to the occurrence frequency of this label in its memory and sends the selected label to the listener.
   c. The listener adds the most popular label received to its memory.
3) Finally, the post-processing based on the labels in the memories and the threshold r is applied to output the communities.

Note that SLPA starts with each node being in its own community (a total of n), the algorithm explores the network and outputs the desired number of communities in the end. As such, the number of communities is not required as an input. Due to the step c, the size of memory increases by one for each node at each step. SLPA reduces to LPA when the size of memory is limited to one and the stop criterion is convergence of all labels. Empirically, SLPA produces relatively stable outputs, independent of network size or structure, when T is greater than 20. Although SLPA is non-deterministic due to the random selection and ties, it performs well on average as shown in later sections.

Post-processing and Community Detection: In SLPA, the detection of communities is performed when the stored information is post-processed. Given the memory of a node, SLPA converts it into a probability distribution of labels. Since labels represent community id's, this distribution naturally defines the

strength of association to communities to which the node belongs. To produce *crisp* communities in which the membership of a node to a given community is *binary*, i.e., either a node is in a community or not, a simple thresholding procedure is performed: if the probability of seeing a particular label during the whole process is less than a given threshold $r \in [0, 0.5]$, this label is deleted. After thresholding, connected nodes having a particular label are grouped together and form a community. If a node contains multiple labels, it belongs to more than one community and is called an *overlapping node*. A smaller value of r produces a larger number of communities. However, the effective range is typically narrow in practice. When $r \geq 0.5$, SLPA outputs disjoint communities.

Complexity: The initialization of labels requires $O(n)$, where n is the total number of nodes. The outer loop is controlled by the user defined maximum iteration T, which is a small constant which in our experiments was set to 100. The inner loop is controlled by n. Each operation of the inner loop executes one speaking rule and one listening rule. The speaking rule requires exactly $O(1)$ operation. The listening rule takes $O(\overline{k})$ on average, where $\overline{k}$ is the average node degree. In the post-processing, the thresholding operation requires $O(Tn)$ operations since each node has a memory of size T. In summary, the time complexity of the entire algorithm is $O(Tn\overline{k})$ or $O(Tm)$, linear with the total number of edges m. The execution times for synthetic networks where averaged for each $\overline{k}$ over networks with different structures, i.e., different degree and community size distributions (see Section 4.1 for details). The results shown in Fig. 1 confirm the *linear* scaling of the execution times. On a desktop with 2.80GHz CPU, SLPA took about six minutes to run over a two million nodes Amazon co-purchasing, which is ten times faster than GCE running over the same network.

4 Tests in Synthetic Networks

4.1 Methodology

To study the behavior of SLPA, we conducted extensive experiments in both synthetic and real-world networks. For synthetic random networks, we adopted the widely used LFR benchmark[2] [8], which allows heterogeneous distributions of node degrees and community sizes.

Table 1. Algorithms in the tests

Algorithm	Complexity	Imp
CFinder [11], 2005	-	C++
LFM [7], 2009	$O(n^2)$	C++
EAGLE [14], 2009	$O(n^2 s)$	C++
CIS [6], 2009	$O(n^2)$	C++
GCE [9], 2010	$O(mh)$	C++
COPRA [5], 2010	$O(vm \log(vm/n))$	Java
NMF [12], 2010	$O(Kn^2)$	Matlab
Link [1], 2010	$O(nk_{max}^2)$	C++
SLPA, 2011	$O(Tm)$	C++

Table 2. The ranking of algorithms

Rank	RS_{Omega}	RS_{NMI}	RS_F
1	SLPA	SLPA	SLPA
2	COPRA	GCE	CFinder
3	GCE	NMF	COPRA
4	CIS	CIS	Link
5	NMF	LFM	LFM
6	LFM	COPRA	CIS
7	CFinder	CFinder	GCE
8	Link	EAGLE	EAGLE
9	EAGLE	Link	NMF

[2] http://sites.google.com/site/andrealancichinetti/files

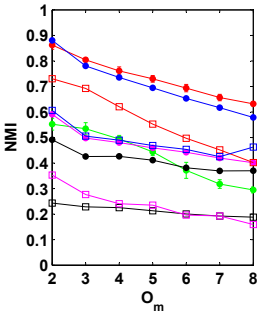


Fig. 2. NMI as a function of the number of memberships O_m in LFR

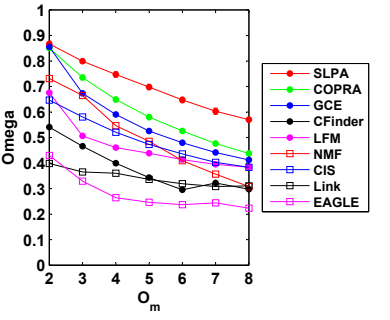


Fig. 3. Omega as a function of the number of memberships O_m in LFR

We used networks with size $n = 5000$. The average degree is kept at $\overline{k} = 10$. The degree of overlapping is determined by two parameters. $\boldsymbol{O_n}$ defines the number of overlapping nodes and is set to 10% of all nodes. $\boldsymbol{O_m}$ defines the number of communities to which each overlapping node belongs and varies from 2 to 8 indicating the diversity of overlap. By increasing the value of O_m, we create harder detection tasks. Other parameters are as follows: node degrees and community sizes are governed by the power laws with exponents 2 and 1; the maximum degree is 50; the community size varies from 20 to 100; the expected fraction of links of a node connecting it to other communities, called the mixing parameter μ, is set to 0.3. We generated ten instance networks for each setting.

In Table 1, we compared SLPA with eight other algorithms representing different categories discussed in section 2. For algorithms with tunable parameters, the performance with the optimal parameter is reported. For CFinder, k varies from 3 to 10; for COPRA, v varies from 1 to 10; for LFM α is set to 1.0 [7]. For *Link*, the threshold varies from 0.1 to 0.9 with an interval 0.1. For SLPA, the number of iterations T is set to 100 and r varies from 0.01 to 0.1. The average performance together with error bar over ten repetitions are reported for SLPA and COPRA. For NMF, we applied a threshold varying from 0.05 to 0.5 with an interval 0.05 to convert it to a crisp clustering.

To summarize the vast amount of comparison results and provide a measure of relative performance, we proposed $RS_M(i)$, the averaged ranking for algorithm i with respect to measure M as follows:

$$RS_M(i) = \sum_{j=1} w_j \cdot rank(i, O_m^j), \tag{1}$$

where O_m^j is the number of memberships in $\{2, 3, \cdots, 8\}$, w_j is the weight, and function *rank* returns the ranking of algorithm i for the given O_m. For simplicity, we assume equal weights over different O_m's in this paper. Sorting RS_M in increasing order gives the final ranking among algorithms.

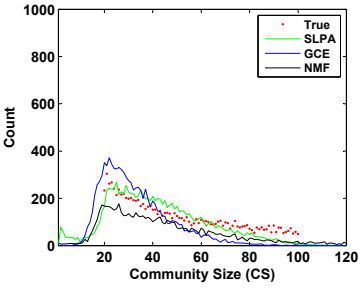


Fig. 4. The unimodal histogram of the detected community sizes for SLPA, GCE and NMF

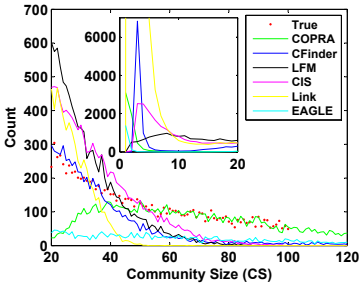


Fig. 5. The bimodal histogram of the detected community sizes for COPRA, CIS, LFM, Link, EAGLE and CFinder

4.2 Identifying Overlapping Communities in LFR

The extended normalized mutual information (NMI) [7] and Omega Index [3] are used to quantify the quality of communities discovered by an algorithm. NMI measures the fraction of nodes in agreement in two covers, while Omega is based on pairs of nodes. Both NMI and Omega yield the values between 0 and 1. The closer this value is to 1, the better the performance is.

As shown in Fig. 2 and Fig. 3, some algorithms behave differently under different measures (the rankings of RS_{Omega} and RS_{NMI} among algorithms in Table 2 also change). As opposed to NMF and COPRA, which are especially sensitive to the measure, SLPA is remarkably stable in NMI and Omega.

Comparing the detected and known numbers of communities and distributions of community sizes (CS) helps to understand the results. On one hand, we expect the community size to follow a power law with exponent 1 and to range from 20 to 100 by design. As shown in Fig. 4, high-ranking (with high NMI) algorithms such as SLPA, GCE and NMF typically yield a *unimodal* distribution with a peak at $CS = 20$ fitting well with the ground truth distribution. In contrast, algorithms in Fig. 5 typically produce a *bimodal* distribution. The existence of an extra dominant mode for CS ranging from 1 to 5 results in a significant number of small size communities in CFinder, LFM, COPRA, Link and CIS. These observations nicely explain the ranking with respect to NMI.

4.3 Identifying Overlapping Nodes in LFR

Identifying nodes overlapping multiple communities is an essential component of measuring the quality of a detection algorithm. However, the node level evaluation was often neglected. Here we first look at the number of detected overlapping nodes O_n^d (see Fig. 6) and detected memberships O_m^d (see Fig. 7) relative to the ground truth O_n and O_m, based on the information in Fig. 2. Note that a value close to 1 indicates closeness to the ground truth, and values over 1 are possible

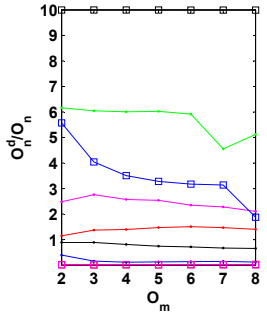


Fig. 6. The number of detected overlapping nodes relative to the ground truth

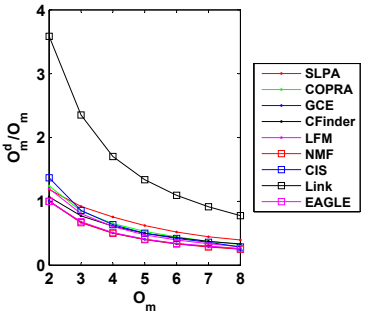


Fig. 7. The number of memberships of detected overlapping nodes relative to the ground truth

when an algorithm detects more nodes or memberships than there are known to exist. As shown, SLPA yields the numbers that are close to the ground truth in both cases.

Note that O_n^d alone is insufficient to accurately quantify the detection performance, as it contains both true and false positive. To provide precise analysis, we consider the identification of overlapping nodes as a *binary classification* problem. A node is labeled as *overlapping* as long as $O_m{>}1$ or $O_m^d{>}1$ and labeled as *non-overlapping* otherwise. Within this framework, we can use F-score as a measure of detection accuracy defined as

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall}, \tag{2}$$

where *recall* is the number of correctly detected overlapping nodes divided by O_n, and *precision* is the number of correctly detected overlapping nodes divided by O_n^d. F-score reaches its best value at 1 and worst score at 0.

As shown in Fig. 8, SLPA achieves the best score on this metric. This score has a positive correlation with O_m while scores of other algorithms are negatively correlated with it. SLPA correctly uncovers a reasonable fraction of overlapping nodes even when those nodes belong to many groups (as demonstrated by the high precision and recall in Fig. 9 and Fig. 10). Other algorithms that fail to have a good balance between precision and recall result in low F-score, especially for EAGLE and Link. The high precision of EAGLE (also CFinder and GCE for $O_m = 2$) shows that clique-like assumption of communities may help to identify overlapping nodes. However, they under-detect the number of such nodes.

With the F-score ranking, GCE and NMF no longer rank in the top three algorithms, while SLPA stays there. Taking both community level performance (NMI and Omega) and node level performance (F-score) into account, we conclude that SLPA performs well in the LFR benchmarks.

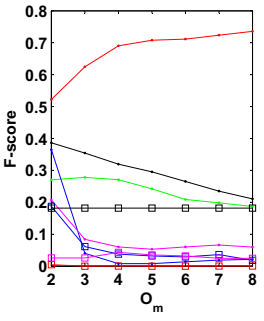


Fig. 8. The F-score

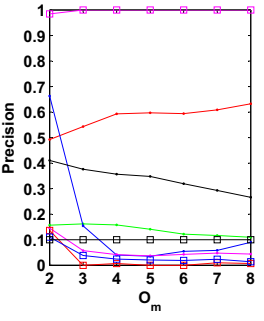


Fig. 9. The precision

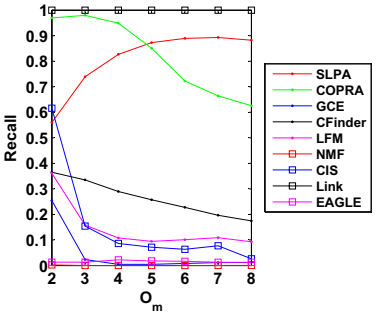


Fig. 10. The recall

Table 3. Social networks in the tests

Network	n	$\overline{k}$	Network	n	$\overline{k}$
karate (KR)	34	4.5	Email (EM)	33696	10.7
football (FB)	115	10.6	P2P	62561	2.4
lesmis (LS)	77	6.6	Epinions (EP)	75877	10.6
dolphins (DP)	62	5.1	Amazon (AM)	262111	6.8
CA-GrQc (CA)	4730	5.6	HighSchool (HS1)	69	6.3
PGP	10680	4.5	HighSchool (HS2)	612	8.0

5 Tests in Real-World Social Networks

We applied SLPA to a wide range of well-known social networks[3] as listed in Table 3. The high school friendship networks that were analyzed in a project funded by the National Institute of Child Health and Human Development, are social networks in high schools self-reported by students together with their grades, races and sexes. We used these additional attributes for verification.

5.1 Identifying Overlapping Communities in Social Networks

To quantify the performance, we used the overlapping modularity, Q_{ov}^{Ni} (with values between 0 and 1), proposed by Nicosia [10], which is an extension of Newman's modularity. A high value indicates a significant overlapping community structure relative to the null model. We removed CFinder, EAGLE and NMF from the test because of either their memory or their computation inefficiency on large networks. As an additional reference, we added the disjoint detection results with the Infomap algorithm.

As shown in Fig. 11, in general, SLPA achieves the highest average Q_{ov}^{Ni}, followed by LFM and COPRA, even though the performance of SLPA has larger fluctuation than that in synthetic networks. Compared with COPRA, SLPA is more stable as evidenced by smaller deviation of its Q_{ov}^{Ni} score. In contrast,

[3] www-personal.umich.edu/~mejn/netdata/ and snap.stanford.edu/data/

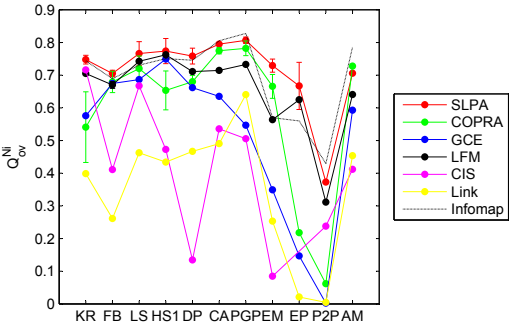


Fig. 11. Overlapping modularity Q_{ov}^{Ni}

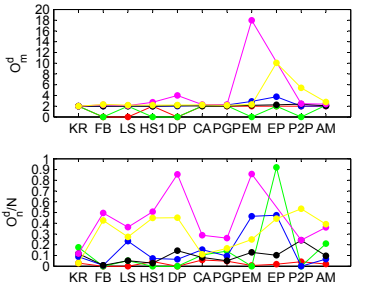


Fig. 12. The number of detected memberships (top) and the fraction of detected overlapping nodes (bottom)

COPRA does not work well on highly sparse networks such as *P2P*, for which COPRA finds merely one single giant community. COPRA also fails on *Epinions* network because it claims too many overlapping nodes in view of consensus of other algorithms as seen in the bottom of Fig. 12. Such over-detection also applies to CIS and Link, resulting in low Q_{ov}^{Ni} scores for these two algorithms. The results in Fig. 12 (based on the clustering with the best Q_{ov}^{Ni}) show a common feature in the tested real-world networks, which is a relatively little agreement between results of different algorithms, i.e., the relatively small overlap in both the fraction of overlapping nodes (typically less than 30%) and the number of communities of which an overlapping node is a member (typically 2 or 3).

As known, a high modularity might not necessarily result in a *true* partitioning as it does in the disjoint community detection. We used the high school network (HS1) with known attributes to verify the output of SLPA. As shown in Fig. 13, there is a good agreement between the found and known partitions in term of student's *grades*. In SLPA, the grade 9 community is further divided into two subgroups. The larger group contains only white students, while the smaller group demonstrates *race* diversity. These two groups are connected partially via an overlapping node. It is also clear that overlapping nodes only exist on the boundaries of communities. A few overlapping nodes are assigned to three communities, while the others are assigned to two communities (i.e., their O_m is 2).

5.2 Identifying Overlapping Communities in Bipartite Networks

Discovering communities in bipartite networks is important because they provide a natural representation of many social networks. One example is the online tagging system with both users and tags. Unlike the original LPA algorithm, which performs poorly on bipartite networks, SLPA works well on this kind of networks. We demonstrate this using two real-world networks[4]. One is a Facebook-like social network. One type of nodes represents users (abbr. FB-M1), while the other

[4] Data are available at http://toreopsahl.com/datasets/

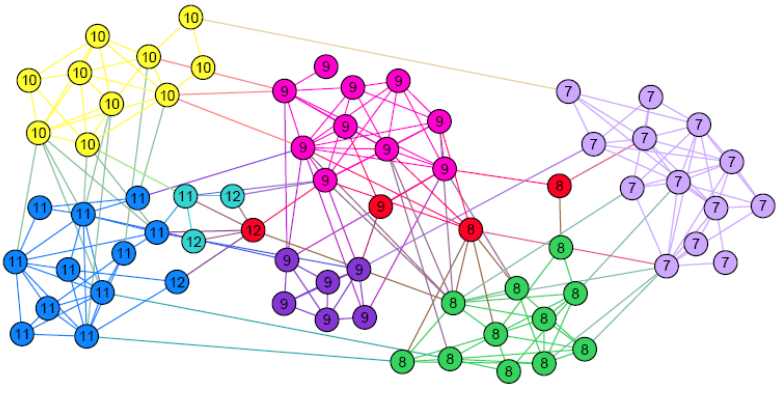


Fig. 13. High school network ($n = 69$, $\overline{k} = 6.4$). Labels are the known grades ranging from 7 to 12. Colors represent communities discovered by SLPA. The overlapping nodes are highlighted by red color.

represents messages (abbr. FB-M2). The second network is the interlocking directorate. One type of nodes represents affiliations (abbr. IL-M1), while the other individuals (abbr. IL-M2).

We compared SLPA with COPRA in Table 4. One difference between SLPA and COPRA is that SLPA applies to the *entire* bipartite network directly, while COPRA is applied to each type of nodes *alternatively*. Q_{ov}^{Ni} is computed on the *projection* of each type of nodes. Again, we allow *overlapping* between communities. Although COPRA is slightly better (by 0.03) than SLPA on the second type of nodes for interlock network, it is much worse (by 0.11) on the first type. Moreover, COPRA fails to detect meaningful communities in the Facebook-like network, while SLPA demonstrates relatively good performance.

Table 4. The Q_{ov}^{Ni} of SLPA and COPRA for two bipartite networks

Network	n	SLPA (std)	COPRA (std)
FB-M1	899	0.23 (0.10)	0.02 (0.07)
FB-M2	522	0.36 (0.02)	0.02 (0.07)
IL-M1	239	0.59 (0.02)	0.48 (0.02)
IL-M2	923	0.69 (0.01)	0.72 (0.01)

5.3 Identifying Overlapping Nested Communities

In the above experiments, we applied a post-processing to remove subset communities from the raw output of stages 1 and 2 of SLPA. This may not be necessary for some applications. Here, we show that rich *nested* structure can be recovered in the high school network (HS2) with $n = 612$. The hierarchy is shown as a treemap[5] shown in Fig. 14. To evaluate the degree to which a discovered community matches the known attributes, we define a *matching score* as the *largest*

[5] Treemap is used for visualization: www.cs.umd.edu/hcil/treemap/

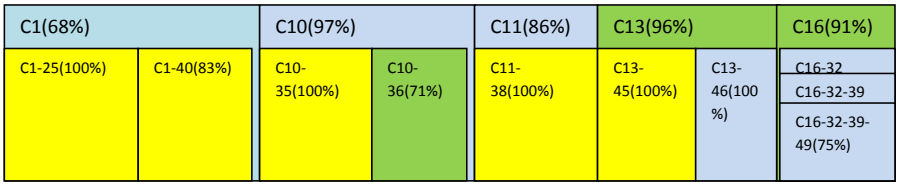


Fig. 14. The nested structure in the high school network represented as a Treemap. The color represents the best explaining attribute: *blue* for *grade*; *green* for *race*; and *yellow* for *sex*. Numbers in parenthesis are the matching scores defined in the text. The size of shapes is proportional to the community size. Due to the page limit, only part of the entire treemap is shown.

fraction of matched nodes relative to the community size among three attributes (i.e., grade, race and sex). The corresponding attribute is said to best explain the community found by SLPA.

As shown, SLPA discovers a tree with a height of four. Most of the communities are distributed on the first two levels. The community name shows the full hierarchy path (connected by a dash '-') leading to this community. For example, *C1* has id 1 and is located on the first level, while *C1-25* has id 25, and it is the second level sub-community of community with id 1.

Nested structures are found across different attributes. For example, *C13* is best explained by *race*, while its two sub-communities perfectly account for *grade* and *sex* respectively. In *C1*, sub-communities explained by the same attribute account for *different* attribute *values*. For example, both *C1-25* and *C1-40* are identified by *sex*. However, the former contains only *male* students, while in the latter *female* students are the majority. Although the treemap is not capable of displaying overlaps between communities, the nested structures overlap as before.

6 Conclusions

We introduced a dynamic interaction process, SLPA as a basis for an efficient and effective *unified* overlapping community detection algorithm. SLPA allows us to analyze different kinds of community structures, such as disjoint communities, individual overlapping nodes, overlapping communities and overlapping nested hierarchy in both unipartite and bipartite topologies. Its underlying process can be easily modified to accommodate other types of networks (e.g., k-partite graphs). In the future work, we plan to apply SLPA to temporal community detection.

Acknowledgments. Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053 and by the Office of Naval Research Grant No. N00014-09-1-0607. The views and conclusions contained in this document are those of the authors and

should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the Office of Naval Research or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

1. Ahn, Y.Y., Bagrow, J.P., Lehmann, S.: Link communities reveal multiscale complexity in networks. Nature 466, 761–764 (2010)
2. Baumes, J., Goldberg, M., Magdon-Ismail, M.: Efficient Identification of Overlapping Communities. In: Kantor, P., Muresan, G., Roberts, F., Zeng, D.D., Wang, F.-Y., Chen, H., Merkle, R.C. (eds.) ISI 2005. LNCS, vol. 3495, pp. 27–36. Springer, Heidelberg (2005)
3. Collins, L.M., Dent, C.W.: Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions. MBR 23, 231–242 (1988)
4. Farkas, I., Ábel, D., Palla, G., Vicsek, T.: Weighted network modules. New Journal of Physics 9(6), 180 (2007)
5. Gregory, S.: Finding overlapping communities in networks by label propagation. New J. Phys. 12, 103018 (2010)
6. Kelley, S.: The existence and discovery of overlapping communities in large-scale networks. Ph.D. thesis, Rensselaer Polytechnic Institute, Troy, NY (2009)
7. Lancichinetti, A., Fortunato, S., Kertesz, J.: Detecting the overlapping and hierarchical community structure in complex networks. New J. Phys., 033015 (2009)
8. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E 78, 046110 (2008)
9. Lee, C., Reid, F., McDaid, A., Hurley, N.: Detecting highly overlapping community structure by greedy clique expansion. In: snakdd. pp. 33–42 (2010)
10. Nicosia, V., Mangioni, G., Carchiolo, V., Malgeri, M.: Extending the definition of modularity to directed graphs with overlapping communities. J. Stat. Mech., 03024 (2009)
11. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature, 814–818 (2005)
12. Psorakis, I., Roberts, S., Ebden, M., Sheldon, B.: Overlapping community detection using bayesian non-negative matrix factorization. Phys. Rev. E 83, 066114 (2011)
13. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E 76, 036106 (2007)
14. Shen, H., Cheng, X., Cai, K., Hu, M.B.: Detect overlapping and hierarchical community structure. Physica A 388, 1706 (2009)
15. Xie, J., Szymanski, B.K.: Community detection using a neighborhood strength driven label propagation algorithm. In: IEEE NSW 2011, pp. 188–195 (2011)
16. Zhang, S., Wang, R.S., Zhang, X.S.: Identification of overlapping community structure in complex networks using fuzzy c-means clustering. Physica A 374, 483–490 (2007)

WeightTransmitter: Weighted Association Rule Mining Using Landmark Weights

Yun Sing Koh[1], Russel Pears[2], and Gillian Dobbie[1]

[1] Department of Computer Science, University of Auckland, New Zealand
{ykoh,gill}@cs.auckland.ac.nz
[2] School of Computing and Mathematical Sciences, AUT University, New Zealand
rpears@aut.ac.nz

Abstract. Weighted Association Rule Mining (WARM) is a technique that is commonly used to overcome the well-known limitations of the classical Association Rule Mining approach. The assignment of high weights to important items enables rules that express relationships between high weight items to be ranked ahead of rules that only feature less important items. Most previous research to weight assignment has used subjective measures to assign weights and are reliant on domain specific information. Whilst there have been a few approaches that automatically deduce weights from patterns of interaction between items, none of them take advantage of the situation where weights of only a subset of items are known in advance. We propose a model, WeightTransmitter, that interpolates the unknown weights from a known subset of weights.

Keywords: Weight Estimation, Landmark Weights, Association Rule Mining.

1 Introduction

Weighted Association Rule Mining has been proposed as a method of generating a compact rule base whose rules contain items that are of most interest to the user [2,8,10,11]. Items are typically weighted based on background domain knowledge. For example, items in a market basket dataset may be weighted based on the profit they generate. However, in many applications pre-assignment of weights is not practical. In high dimensional datasets containing thousands of different items it may not be feasible to gather domain specific information on every single item, especially in a dynamically changing environment. In such situations it is more practical to exploit domain information to set weights for only a small subset of items (which we refer to as landmark items) and to then estimate the weights of the rest through the use of a suitable interpolation mechanism. This research addresses the issue of constructing a suitable model which will facilitate the estimation of unknown weights in terms of a given small subset of items with known weights.

Another key issue that needs to be addressed is the validity of assigning item weights based on domain specific input alone. Typically, items are supplied weights based on their perceived importance, for example the profit that they generate. However, such weight assessments are made in isolation to other items and thus do not account for the indirect profit that an item generates by promoting the sale of other items which may

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 37–48, 2012.

be of high profit. For example, retailers often reduce their profit margin on items that already have relatively low profit and market them as a package deal involving high profit items. A concrete example is a discount on a mobile handset that is conditional on the customer signing a long term contract with the phone company involved. In such situations, the "low profit" item (mobile handset) is used as an incentive to entice customers into buying the high profit items (calling plan contract). Clearly, in such contexts the actual profit margin of the low profit item does not accurately reflect its importance. Thus one of the premises of this research is that domain input on item weighting even when available may not be sufficient by itself in characterizing the importance of an item. Transactional linkages between items add value to domain specific information and when these two inputs are combined in a suitable manner a more accurate assessment can be made on an item's importance.

Two major contributions of this research are the development of a model that expresses the unknown weights of items in terms of known weights (landmark weights) and an interpolation mechanism that estimates weights by taking into account linkages between items that occur together. The rest of the paper is organized as follows. In the next section, we examine previous work in the area of weighted association rule mining. In Section 3 we give a formal definition of the weighted estimation problem. Section 4 presents our model for weight estimation. Our experimental results are presented in Section 5. Finally we summarize our research contributions in Section 6.

2 Related Work

In the context of weighted association rule mining a number of different schemes have been proposed for item weighting. Most of the schemes propose that domain information be utilized for setting weights for items. Tao et al. [9], Cai et al. [2] and Sanjay et al. [6] propose that item profit be utilized for assigning weights in retail environments for items while Yan et al. [11] use page dwelling time to assign weights in a web click stream data environment. More recent work reported in [8,3,4] took a different approach to the weight assignment problem. Sun and Bai introduced the concept of w-support which assigns weights to items based on the properties of transactions in a given dataset thus removing the need for domain specific input. The dataset was first converted into a bipartite graph, with one set of nodes representing the items and the other set of nodes representing the transactions. The w-support was then calculated by counting the links between items and the transactions that the items appeared in. Koh et al. [3] proposed a Valency model where the weight of an item was defined as a linear combination of its *purity* and its *connectivity*. Purity takes into account the number of items that a given item interacts with, while Connectivity accounted for the degree of interaction between an item and its neighboring items. Pears et al. [4] used a weight inference mechanism based on Eigenvectors derived from a transactional dataset. Given a dataset D, the covariance of every pair of items that occur in transactions across D was expressed in terms of its covariance matrix M. The first Eigenvector derived from the covariance matrix M was used to assign weights to items.

None of the work done so far in weight inference directly addresses the issue of weight estimation from a set of known landmark weights. A simple extension such as

restricting the set of items input to only include the unknown items will not suffice as the weights will be computed only on the basis of the interactions between the set of unknown items and the interactions with the landmark items will be neglected. As such, none of the above work can directly be utilized in their entirety.

3 Problem Definition

The weight fitting problem that we frame is to estimate the *overall* weight of items and not simply the domain specific weights for items that are unspecified (*i.e.*, item not in the landmark set, L). As stated in the introduction, certain items that are perceived to be of low importance on the basis of domain knowledge may actually assume a higher importance than their perceived rating due to strong interactions with items that are of high importance. In our problem setting we associate with each item a domain weight and an interaction weight. Domain weights dw are only available for the set of landmark items, L, whereas interaction weights iw are available for *all* items as these can be deduced from the co-occurrences of items given a transaction database.

Given a set of items I, a subset L of landmark items where $L \subset I$, and a transaction database D, the acquired weight w_i of a given item i is determined by:

$$w_i = \frac{\sum_{l \in L} iw(i,l).(w_l + dw_l) + \sum_{m \in M} iw(i,m).(w_m)}{\sum_{k \in N} iw(i,k)} \quad (1)$$

where N represents the set of neighbors of item i, $dw_i \geq 0$ when $i \in L$ and $dw_i = 0, otherwise$. Thus an item i acquires a weight from its interactions with its neighbors who transmit their own weights in a quantity proportional to the degree of interaction, iw. Neighbors that are landmarks transmit their domain weights as well as their acquired weights while neighbors in the set M of items that are not landmarks only transmit their acquired weights. In the context of this research a neighbor of a given item i is taken to be any item j that co-occurs with item i when taken across the database D. In effect, an item that is a landmark item contributes both its own domain weight and the weight acquired from its neighbors, while non landmark items simply transmit their acquired weight which in turn was obtained from their own interactions with neighboring items, which could include landmark items. Henceforth we shall abbreviate the term acquired weight simply by the term weight, except when it is necessary to emphasize the composite nature of the weight assignment.

The accuracy of the weight estimation mechanism expressed by Equation 1 above is dependent on how the interaction component is modeled. This specification is a modeling issue which does not impact on the general definition of the problem and so further discussion of this component is deferred to Section 4 which deals with the model developed to solve the problem.

For a given set of landmark items L the problem can now be stated formally as follows: return all items $i \in H$ where

$$H = \{i | i \in I \text{ is in the top } p\% \text{ of items when ranked on acquired weight from Eq (1)}\} \quad (2)$$

where p is a user-supplied threshold that determines the minimum overall weight to be returned to the user for use in a subsequent weighted association rule mining phase.

4 Weight Transmitter Model

In this section we present a model that we use as the basis of the solution to the weight estimation problem. We use a graph structure (N, E) where nodes are represented by items and edges by interactions between pairs of items. Each node i is associated with the weight w_i of the item, while an edge between items i and j is represented by $G(i, j)$ where G is the Gini information index [5]. The Gini information index $G(i, j)$ measures the degree of dependence of item j on item i. A high value of $G(i, j)$ indicates that item j occurs with a high degree of probability whenever item i occurs, likewise, the non occurrence of item i leads to the non occurrence of item j with a high degree of probability. Thus the G value captures the degree of dependence between pairs of items. The higher the dependence of item j on item i, the higher the proportion of weight transmission from item i to item j, and as such the Gini index can be used to express the interaction weight component of the model.

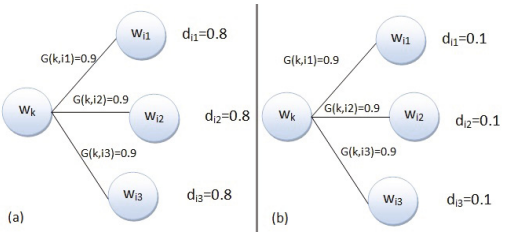


Fig. 1. Influence of Neighborhood in Weight Estimation

As an illustrative example, consider two different scenarios with four items whereby we have item k with unknown domain weight and three other items $i1$, $i2$, and $i3$ with known domain weights. In the first case, (Figure 1 (a)) each of $i1$, $i2$ and $i3$ have domain specified weights of 0.8 and each interacts with item k with a G value of 0.9. The WeightTransmitter model that we propose returns a weight value of 2.4 for each of the items, which when normalized yields a value of 0.89. With the same set of items but with domain specific weights set to 0.1 (Figure 1(b)) all weights for the four items end up with the same value of 0.3, which when normalized yields a value of 0.11. This example illustrates the importance of neighborhood in the weight estimation process; an item which is strongly connected through high G values to high weight items will acquire a high weight, whereas the same item when connected to low weight items will acquire a low weight, regardless of the strength of the connections.

We now present the WeightTransmitter model by expressing the weight of a given item k in terms of the weights of its neighbors as:

$$w_k = \frac{\sum_{i \in S_1} G(i,k).(w_i + dw_i) + \sum_{j \in S_2} G(j,k).(w_j)}{\sum_{i \in S_1} G(i,k) + \sum_{i \in S_2} G(i,k)} \tag{3}$$

where S_1 represents the set of neighbors of item i whose domain supplied weight dw_i components are known in advance and S_2 is the set of neighbors of item i whose domain

weights are unknown. Now $\sum_{i \in S_1} G(k,i) + \sum_{i \in S_2} G(k,i)$ represents a known quantity c_{1k}, since all G index values can be calculated from the transaction database. The dw_i terms in set S_1 also represent known quantities. We denote $\sum_{j \in S_1} G(k,i).dw_i$ by c_{2k}. Substituting the known constants c_{1k}, c_{2k} in the above equation and re-arranging the terms gives:

$$c_{1k}.w_k - \sum_{i \in S} G(i,k).w_i = c_{2k} \tag{4}$$

where $S = S_1 \cup S_2$ represents the complete neighborhood of item k. The above now represents a system of k linear simultaneous equations in k unknowns which has an exact solution with the Gaussian elimination method which we employ. The algorithm below illustrates how the WeightTransmitter model fits in with the traditional weighted association rule mining algorithm.

Algorithm: WeightTransmitter Model
Input: Transaction database T, known landmark weights dw, universe of items I
Output: Item Weights W

Step 1: Build a one level graph of the neighborhood of item i
$N(i) \leftarrow \{k | k \in t, t \in T, i \in t\}$
Step 2: Calculate G values for interactions between item i and neighbors $N(i)$
Step 3: Compute $C1 = \{|c_{1k}k \in I\}$ and $C2 = \{|c_{2k}k \in I\}$
Step 4: Solve for weight vector W
$W \leftarrow \{w(i) | \text{GaussianElimination}(I, C1, C2), i \in I\}$

5 Experimental Results

Our evaluation is divided into three sections: weight estimation evaluation, rule evaluation, and runtime evaluation. In the next section we describe the datasets that were used in these evaluations.

5.1 Datasets

Our experiments were conducted on five real-world datasets which are described below.

- **Retail dataset.** We used a retail market basket dataset supplied by a supermarket store that contained the unit profit values for each item which were supplied in a separate file [1].
- **Nasa weblog datasets.** We also used two different web log files from the NASA Kennedy Space Center in Florida collected over the months of July and August 1995. In these datasets pages represented items, and transactions consisted of a sequence of clicks on a set of web pages that took place across a session, which we set to have a maximum time of 15 minutes. The average dwelling time on a web page (taken across all transactions) was taken as a proxy for item weight.
- **Computer Science Lab datasets.** Finally, we used two datasets containing web log requests from a computer science lab at the University of Auckland between the months of December 2007 - February 2008, and February 2008 - December

2008. We preprocessed the dataset using the same technique as the Nasa datasets and used the same proxy for item weight. Overall there were 1764 items and 5415 instances for the first of these datasets, while the second had 2315 items and 5591 instances.

5.2 Weight Estimation Evaluation

This evaluation was designed with three key objectives in mind. Firstly, to establish the degree of sampling required in order to achieve convergence between estimated and actual weight on the composite weight measure. Ideally, convergence should be achieved at a low level of sampling for the weight estimation process to be effective. Secondly, to identify items that were flagged as being low weight according to domain information but were assigned high weight values by the WeightTransmitter model. These items are potentially interesting as they highlight situations where domain knowledge is inadequate to characterize the true importance of such items. Thirdly, we wanted to assess the level of accuracy achieved by the weight estimation process at the point of convergence. Since we had access to the domain weights for the complete set of items we were able to establish the ground truth in terms of the composite weights by simply running WeightTransmitter with a landmark sampling level at 100%.

To evaluate the accuracy of the weights produced by the WeightTransmitter model we varied the percentage of landmark weights in the range 10% to 90% and tracked the overall accuracy and precision across the high weight items. We start with the accuracy evaluation. At each of the sampling levels 30 different runs were used that chose different sets of landmark items at random. The accuracy measures presented represent an average of the measure taken across the 30 different trials.

Weight Convergence and Accuracy Analysis. Accuracy was tracked using three different measures: Correlation, Precision on high weight items, and Target Lift [7]. Target Lift is a measure commonly used to measure the lift in response rate that occurs when marketing offers are sent to a small set of customers who are likely to respond (identified through some prediction method) rather than a mass marketing campaign that targets the entire population. In the context of weight estimation the set of items returned by WeightTransmitter which it regards as high weight corresponds to the set of probable customers and the universe of items represents the entire customer population.

In the first analysis, we ran WeightTransmitter with the set of landmark weights as input and collected the results into the set S_l. We then re-ran WeightTransmitter with the complete set of known weights as input and collected the results into the set S_c. We then plotted the Pearson correlation between the two result sets against the sampling percentages that we used for the landmark weights. Figure 2 shows that there is a stabilization of correlation around the 30% mark; the average correlation value is 89%, with a standard deviation of 0.07. As expected, as the percentage of landmark items increases the greater is the degree of convergence between estimated weight and actual values on the composite weight value. Figure 2 shows that reasonable convergence of weights is achieved around the 30% mark.

For the second analysis each of the sets S_l and S_c were divided into two parts (bins): *low* and *high*. For each of the two sets, the top 10 percentile of items in terms of weight

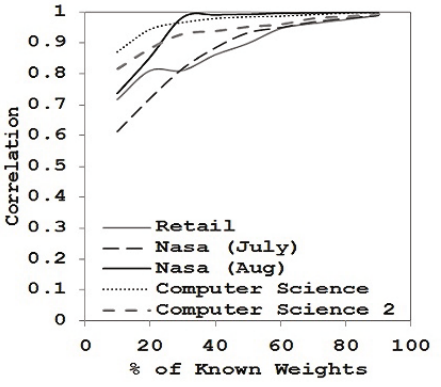


Fig. 2. Correlation Analysis

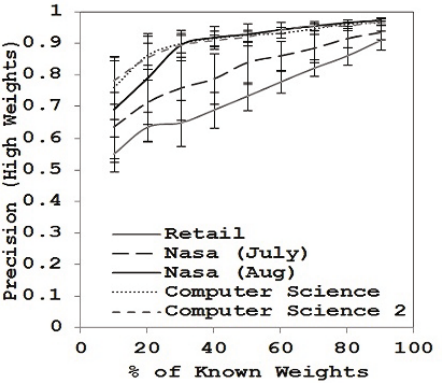


Fig. 3. Precision Analysis (High Weights)

were allocated to *high* bin, and all other items to the *low* bin. Using the bins based on the set S_c(*i.e.*, by running WeightTransmitter at 100% level of sampling) to establish a benchmark we were able to compute precision on the high weight category (bin). Figure 3 shows the precision in the high weight weight category as a function of the sampling percentage. At 30% the average precision value for the high weight items is 80%, with a standard deviation of 0.06.

In the third analysis we calculated the target lift. Table 1 shows that the lift in the true positive rate at a 30% sampling rate is much greater than 1 across all datasets, thus demonstrating the effectiveness of WeightTransmitter over a random weight assignment scheme in identifying high weight items.

Table 1. Target Lift Value at 30%

Dataset	Retail	Nasa (July)	Nasa (Aug)	Computer Science	Computer Science 2
Target Lift	5.04	6.99	7.75	8.53	3.41

Profit Analysis. Our weight accuracy analysis in the previous section establishes the effectiveness of our model in accurately estimating composite weight. However, we were also interested in tracking our other research premise which was the effect of the weighting scheme on items that interacted strongly with items that were known to have high weight. In particular, we were interested in tracking the set of items (H') where $H' = \{i | i \in I$ where i is in the top p percentile on the basis of composite weight but not on the basis of domain weight$\}$. We were able to compute the set H' as we had access to the weights of all items. For all items belonging to H' we defined a profit measure (P) that took into account the amount of indirect profit that such items generated. The profit measure for a given item $i \in H'$ was computed by taking the total profit (P_1) over all transactions (T_1) in which item i occurs and then subtracting from this value the total profit (P_2) over all transactions (T_2) in which item i does not occur.

In order to isolate the confounding effect of transactions in T_2 having more items than T_1 we restricted each of the transactions involved in T_2 to only have the neighbors of the item i under consideration. Furthermore, we also compensated for the differences in the sizes of T_1 and T_2 by scaling P_1 with the factor $\frac{|T_2|}{|T_1|}$.

$$P(i) = \frac{|T_2|}{|T_1|}. \sum_{k \in t_1} \sum_{t_1 \in T_1} w(k) - \sum_{k \in t_2} \sum_{t_2 \in T_2} w(k), \forall t_1, t_2 \in T \tag{5}$$

where $w(k)$ represents the weight of a high weight item k that is connected to item i and T is the set of all transactions in the transaction database. Equation 5 as defined above captures the indirect profit due to item i without the effects of the confounding factors just mentioned. However, the profit measure P by itself has little meaning unless it is compared with the profit generated by the set of items NH that remain low in weight without making the transition to the high weight category. For our premise that domain input on item weighting may not be sufficient by itself in characterizing the importance of an item the P values of items in the set H' needs to be substantially higher than the profit values in the set NH. Table 2 shows that this is indeed the case as the values in the H' column contains much higher values than the NH column for all of the datasets that we tracked. Table 2 contains the following columns; the percentage of items which have transited to the high weight category when transactional linkages are accounted for, average profit of items rated high by WeightTransmitter but not by domain weighting (*i.e.*, the set H'), average profit of items rated high by domain weighting (*i.e.*, the set H''), and average profit of items that were not rated high by WeightTransmitter (*i.e.*, the set NH).

Table 2. Weight Evaluation Based on Profit

Dataset	% Change	H' Items	H'' Items	NH Items
Retail	10	4647.53	3371.64	2418.20
Nasa (July)	11	5448.06	4375.46	3027.62
Nasa (Aug)	11	5101.86	4387.05	3424.50
Computer Science	11	99006.29	57231.93	49504.58
Computer Science 2	11	46219.19	32158.67	40224.14

Sensitivity Analysis. Given that the WeightTransmitter model achieved a high level of precision at the relatively small sampling level of 30% we were interested in investigating how robust the scheme was to changes in the data distribution. In particular, we were interested in tracking the sensitivity of Precision to the degree of variance in the data. Due to the fact that WeightTransmitter uses a sample defined over the set of landmark items, the question that arises is whether the error caused by the sampling remains stable or changes substantially when the underlying data distribution changes. To investigate this issue we used the Retail, Nasa (June), and Computer Science 1 datasets. Each weight value w in each of the selected datasets was perturbed by adding white Gaussian noise to it. Each weight value w for a given set was transformed into a weight value, $w_p = w + N(0, w/d)$, where d is a parameter that controls the level of variance injected into the dataset. We experimented with different values of d so as to obtain 3 levels of drift in variance from the baseline. The drift levels we used

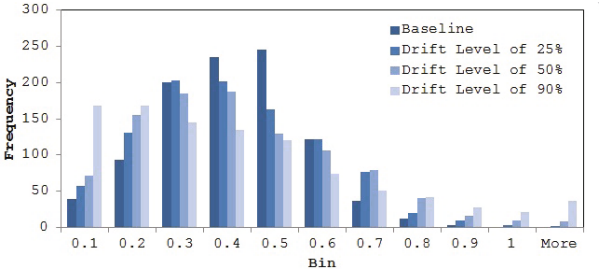


Fig. 4. Histogram of Support Distribution (Computer Science Dataset)

were 25%, 50% and 90% where drift level for a transformed dataset D' is defined by: $drift(D') = \frac{(stdev(D') - stdev(D))}{stdev(D)}$, where $stdev(D), stdev(D')$ represents the standard deviations across the baseline and perturbed datasets respectively. Figure 4 is the histogram of support distribution for the Computer Science dataset. The other datasets follow a similar distribution. The baseline represents the situation when the complete ground truth is known, *i.e.*, the domain weights for all items are known, thus enabling the composite weight to be calculated exactly with no error involved. As mentioned before we had access to the complete set of domain weights for each of the datasets that we experimented with, thus enabling us to measure the true deviation in precision with the degree of drift.

Table 3. Precision results deviation from the baseline

Dataset	Percentages of Known Items									Average
	10%	20%	30%	40%	50%	60%	70%	80%	90%	
Retail 25%	3.74	6.34	4.62	1.21	6.94	1.57	0.16	0.28	0.00	2.76
Retail 50%	3.38	5.43	3.03	2.82	7.02	1.15	2.02	3.31	0.23	3.16
Retail 90%	2.92	7.34	7.97	5.82	9.91	9.56	10.53	10.59	1.01	7.29
Nasa 25%	1.79	3.42	2.39	0.64	1.30	0.25	0.24	0.00	0.12	1.13
Nasa 50%	2.00	1.95	1.68	2.04	0.50	0.19	0.00	0.24	0.24	0.98
Nasa 90%	1.65	0.27	0.71	1.21	1.73	0.50	1.04	0.24	0.12	0.83
CS 25%	2.85	0.00	0.32	1.78	0.31	0.31	0.00	0.00	0.10	0.63
CS 50%	2.09	0.70	0.32	0.21	0.00	0.31	0.93	0.41	0.51	0.61
CS 90%	6.20	1.19	0.00	0.52	0.94	0.00	0.21	0.21	0.21	1.05

Table 3 shows that for the Retail dataset the deviation in Precision from the baseline ranged from 0 to 10.59%. In general as the level of sampling increased the error decreased. The deviation showed some sensitivity to the degree of variance in the data; as the drift level increased the deviation tended to increase. However, even at the extreme drift level of 90%, the deviation was no more than 10%. A similar pattern was observed for the Nasa and Computer Science datasets although the extent of the decrease in precision at the higher degrees of drift was on a smaller scale than with the Retail dataset. These results demonstrate that WeightTransmitter was not overly dependent on which items were chosen as landmarks, even with data that had a very high degree of variability. This is a very desirable feature of a weight estimation mechanism in general and

in terms of WeightTransmitter it inspires more confidence that the good performance at the 30% sampling level will generalize to a wide variety of different datasets.

5.3 Rule Evaluation

One of the major premises behind this research was that the true weight of an item is dependent not just on its individual importance, but also by its interaction with other items that it co-occurs with. For our premise to be true the rule base should contain rules of the form $X \rightarrow Y$ where X is a low weight item based on domain knowledge whereas Y is a highly rated item on the basis of domain knowledge. If such patterns occur then they signify that the set X of items appearing in rule antecedents should be weighted much more heavily than what is suggested on the basis of domain knowledge alone as such items co-occur strongly with highly weighted items.

The rule base was generated by inputing the top $p\%$ of items produced by WeightTransmitter to a standard rule generator. The rules generated for each dataset were subjected to a minimum support threshold of 0.03, confidence threshold of 0.75 and a lift threshold of 1.0. We computed rule interest measures such as Coherence and All Confidence and ranked the rule bases by the Coherence measure. We then analyzed the rule base to look for patterns of the form $X \rightarrow Y$ as described above that either support or refute this premise. The top p parameter was set at 20% for the Retail dataset and at 40% for the rest of the datasets. Figure 5 shows a small sample of 4 rules produced on the Retail dataset that exhibit this pattern.

541145000000 (Low) → 250000 (High)
210000 (Low) 541145000000 (Low) → 250000 (High)
5400136 (Low) → 541015000000 (High)
540014000000 (Low) → 250000 (High)

Fig. 5. Sample of rules

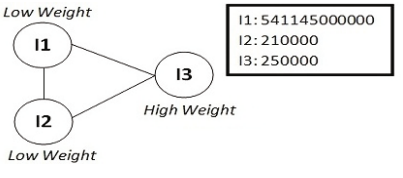


Fig. 6. Sample of WeightTransmitter Model

The presence of such rules validates one of the major premises behind this research. The rule bases produced from the other 3 datasets also exhibited such patterns but could not be presented due to the limitations of space. In terms of the Retail environment the practical value of such rules is that although items such as 541145000000 and 210000 are low profit items they are nevertheless important as the purchase of these items leads to the purchase of the high profit item, 250000. It is also important to note that the rules above would not have been generated if the items were weighted merely on the basis of their domain weights (i.e profit margins) alone as they would have not met the top $p\%$ threshold and would thus not have participated in the rule generation phase. As such, this represents one of the key contributions of this research.

Rules 1 and 2 in Figure 5 reveal the existence of a clique of 3 items: 541145000000, 210000, and 250000 that interact with each other strongly as shown in Figure 6. In the WeightTransmitter model item 250000 transmits its high domain weight to both

items 541145000000 and 210000 in proportions $G(3,1)$ and $G(3,2)$ respectively, thus increasing the domain weights of item 1 (541145000000) and item 2 (210000). This results in transforming these two items into high weight items. At the same time each of items 1 and 2 transmit their respective domain weights to item 3 in proportion to $G(1,3)$ and $G(2,3)$ thus increasing the weight of item 3. This transmission of weights, although increasing the weight of item 3 does not have a significant effect as item 3 is already of high weight.

5.4 Runtime Evaluation

As shown in the previous section the WeightTransmitter model leads to the discovery of valuable knowledge in the form of patterns that can be exploited in a useful manner. However, the model does introduce run time overheads in solving a system of linear equations. As such, our final experiment was to quantify what these overheads were and to ascertain whether the rule generation run time remained within reasonable bounds. Table 4 shows the runtime (measured in seconds) for our experiments with 30%, 60%, and 90% of items used as landmarks, along with the time taken to generate a rule base on the basis of domain knowledge alone, without the use of the WeightTransmitter model. In generating the latter rule base we used exactly the same constraints on minimum support, Confidence and Lift (with the same top p value) in order to keep the comparison fair. Table 4 reveals that the run time overhead introduced by WeightTransmitter does remain within reasonable bounds and that such overhead tends to decrease as a higher rate of landmark sampling is used. The decrease in run time at higher sampling levels is caused by the reduced number of operations required to transform the initial matrix into row echelon form due to the presence of more known values in the form of domain weights. The only result that goes against the above trend was with the Computer Science 2 dataset where the run time actually increased for the generation of the rule base built with the use of domain knowledge only. This was due to the larger number of items being returned in the top p list when compared to the list generated by WeightTransmitter. This resulted in a larger number of itemsets being generated which in turn resulted in a larger rule base, thus contributing to the increase in run time.

Table 4. Execution Time

Dataset	30% Known Weights	60% Known Weights	90% Known Weights	Original Weights
Retail	476	355	234	102
Nasa	56	45	40	14
Nasa Aug	58	48	45	14
Computer Science	48	43	50	45
Computer Science 2	111	108	107	211

6 Conclusions

This research has revealed that weight estimation based on a small set of landmark weights can be performed accurately through the use of the novel WeightTransmitter model that we introduced. Furthermore, we showed through a Profit Analysis conducted

on ground truth data that a substantial percentage of items switched status from the low or moderate weight categories to the high weight category, thus supporting our premise that weight assessments on an item should not be made in isolation to other items.

The use of other methods other than simple random sampling to identify landmark items will be explored in future work. As alternatives to simple random sampling, we plan to investigate the use of stratified random sampling as well as entropy based methods to identify influential items that will act as landmarks.

References

1. Brijs, T., Swinnen, G., Vanhoof, K., Wets, G.: Using association rules for product assortment decisions: A case study. In: Knowledge Discovery and Data Mining, pp. 254–260 (1999)
2. Cai, C.H., Fu, A.W.C., Cheng, C.H., Kwong, W.W.: Mining association rules with weighted items. In: IDEAS 1998: Proceedings of the 1998 International Symposium on Database Engineering & Applications, pp. 68–77. IEEE Computer Society, Washington, DC (1998)
3. Koh, Y.S., Pears, R., Yeap, W.: Valency Based Weighted Association Rule Mining. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010. LNCS, vol. 6118, pp. 274–285. Springer, Heidelberg (2010)
4. Pears, R., Sing Koh, Y., Dobbie, G.: EWGen: Automatic Generation of Item Weights for Weighted Association Rule Mining. In: Cao, L., Feng, Y., Zhong, J. (eds.) ADMA 2010, Part I. LNCS, vol. 6440, pp. 36–47. Springer, Heidelberg (2010)
5. Raileanu, L.E., Stoffel, K.: Theoretical comparison between the gini index and information gain criteria. Annals of Mathematics and Artificial Intelligence 41(1), 77–93 (2004)
6. Ramkumar, G.D., Sanjay, R., Tsur, S.: Weighted association rules: Model and algorithm. In: Proc. Fourth ACM Int'l Conf. Knowledge Discovery and Data Mining (1998)
7. Roiger, R.J., Geatz, M.W.: Data Mining: A Tutorial Based Primer. Addison Edu. Inc. (2003)
8. Sun, K., Bai, F.: Mining weighted association rules without preassigned weights. IEEE Trans. on Knowl. and Data Eng. 20(4), 489–495 (2008)
9. Tao, F., Murtagh, F., Farid, M.: Weighted association rule mining using weighted support and significance framework. In: KDD 2003: Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining, pp. 661–666. ACM, New York (2003)
10. Wang, W., Yang, J., Yu, P.S.: Efficient mining of weighted association rules (WAR). In: KDD 2000: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 270–274. ACM, New York (2000)
11. Yan, L., Li, C.: Incorporating Pageview Weight into an Association-Rule-Based Web Recommendation System. In: Sattar, A., Kang, B.-h. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 577–586. Springer, Heidelberg (2006)

Co-occurring Cluster Mining for Damage Patterns Analysis of a Fuel Cell

Daiki Inaba[1], Ken-ichi Fukui[2], Kazuhisa Sato[3], Junichirou Mizusaki[4], and Masayuki Numao[2]

[1] Graduate School of Information Science and Technology, Osaka University, Japan
[2] The Institute of Scientific and Industrial Research, Osaka University, 8-1 Mihogaoka, Ibaraki, Osaka, 567-0047, Japan
[3] Graduate School of Engineering, Tohoku University, Japan
[4] Institute of Multidisciplinary Research for Advanced Materials, Tohoku University, Japan

Abstract. In this study, we research the mechanical correlations among components of solid oxide fuel cell (SOFC) by analyzing the co-occurrence of acoustic emission (AE) events which are caused by damage. Then we propose a novel method for mining patterns from the numerical data such as AE. The proposed method extracts patterns of two clusters considering co-occurrence between clusters and similarity within each cluster at the same time. In addition, we utilize the dendrogram obtained from hierarchical clustering for reduction of the search space. We applied the proposed method to AE data, and the damage patterns which represent the main mechanical correlations were extracted. We can acquire novel knowledge about damage mechanism of SOFC from the results.

Keywords: clustering, co-occurrence pattern, damage evaluation.

1 Introduction

The fuel cell is regarded as a highly efficient, low-pollution power generation system that produces electricity by direct chemical reaction. However, a crucial issue in putting SOFCs into practical use is the establishment of a technique for evaluating the deterioration of SOFCs in the operating environment. Since SOFCs operate in harsh environments (i.e., high temperature, oxidation and reduction), the reaction area is decreased by fracture damage, and the cell performance is reduced as a result[1]. Two of the co-authors have succeeded in observing mechanical damage to SOFCs using the acoustic emission (AE) method[2]. Acoustic emission is an elastic wave (i.e., vibration, sound waves, including ultrasonic wave) produced by damage, such as cracks in the material, or by friction between materials. Depending on the "fracture mode" (i.e., opening or shear), the type of material, the fracture energy, the shear rate, and other factors, distinct AE wave forms are produced[3,4].

Because AE data is enormous and high dimensional, data mining techniques have been applied for AE data in order to help SOFC experts discover the type

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 49–60, 2012.

and cause of damage. Fukui et al. used kernel self-organizing map (kernel SOM) to succeed in understanding the overview of damage process visually[5]. Also Kitagawa et al. used KeyGraph and density estimation combining with kernel SOM to identify the damage transition and rare essential events[6]. However, little knowledge about the mechanical correlation of damages has been obtained.

Hence, this paper aims to extract damage patterns which represent major mechanical correlation among components in SOFC. For such purpose, this paper proposes a novel method of co-occurrence pattern extraction against numerical data such as AE events. The proposed method determines the area (or the components) of two co-occurring clusters considering co-occurrence between clusters and similarity within each clusters. The experiments show that we can acquire novel knowledge about damage mechanism of SOFC from damage pattern, even for the SOFC experts.

2 The Proposed Method: *Co-occurring Cluster Mining*

2.1 Problems of the Conventional Methods

The task to extract co-occurring AE events is equivalent in some part to the well researched frequent pattern mining. Frequent pattern mining is to extract item sets appering frequently. An item is mainly symbolic data, however, there are also methods such as QLIQUE[8] and mining quantative frequent itemsets[9], which can handle numeric item. In [8] and [9], frequent item sets are extracted by searching frequent subspace clusters. However, the purpose is different from our work, because the above works do not search co-occurrence between clusters.

The straightforward approach to extract co-occurrence patterns from numerical data is first to execute clustering, and then to extract patterns. For example, Honda and Konishi quantized by SOM the image data of clouds obtained from the satellite, and then extracted association rules about the climate change[10]. Also, Yairi et al. extracted association rules about anomaly detection after clustering from time series data transmitted by the satellite[11]. After clustering, the correlation pattern extraction method among clusters are used in these researches, namely, two steps pattern extraction method.

However, the above works do not consider the co-occurrence among clusters during clustering process. As a result, clusters may contain data points which are not related to the co-occurrence patterns. To the contrary, clusters may not contain data points which are related to the co-occurrence patterns. For example, in SOFC, a glass seal changes its state according to the temperature, and the feature of caused AE events also changes gradually. We aim to extract a part of AE events caused by the damage of the glass seal which co-occur with other components of SOFC.

2.2 The Requirements of a Co-occurrence Pattern

In this section, we define the characteristics of data this work handles, then define the requirements of the co-occurrence pattern.

Definition 1 (*numerical event sequence*). Suppose N numerical data $\mathbf{x}_i = (x_{i,1}, \cdots, x_{i,v})$, $(i = 1, \cdots, N)$ in v dimensional space are obtained in order $\mathbf{x}_1, \cdots, \mathbf{x}_N$.

Definition 2 (*basket*). Suppose an event sequence with numerical value is divided into some sections in time series. Namely, let all data set $\mathcal{D}$ is denoted by $\mathcal{D} = [\mathbf{x}_1, \cdots, \mathbf{x}_i][\mathbf{x}_{i+1}, \cdots, \mathbf{x}_j] \cdots [\mathbf{x}_k, \cdots, \mathbf{x}_N]$, $(i < j < k < N)$, where "$[\cdot]$" refers to a basket.

The baskets are given by a minute, a day and so on, besides the length of baskets is not always regular. The details about SOFC data is described in section 3.2.

Here, extracted co-occurrence patterns must satisfy the following three requirements:

Requirement 1 (*correlation*). As for two sets composed of events $A, B \subset \mathcal{D}$, the co-occurring ration of A and B must be high rate.

E.g., Jaccard coefficient, confidence as in association rule, etc.

Requirement 2 (*frequency*). The number of times of which A and B co-occurs in the time series are over the certain number of times.

E.g., support, etc.

Requirement 3 (*similarity*). As for two event sets A and B, events in each event set is similar each other.

E.g., variance in a cluster, average distance between data points in a cluster, etc.

Requirements 1 and 2 are derived from the co-occurrence between event sets (cluster), and requirement 3 is from the clustering of the events.

Definition 3 (*co-occurring cluster*). If event sets with numerical value $A, B \subset \mathcal{D}$ satisfy the above three requirements, set A is a co-occurring cluster of B.

Definition 4 (*co-occurrence pattern*). With co-occurring clusters A and B which satisfy all three requirements, $P(A, B) = \{A, B | A \cap B = \emptyset\}$ is called a co-occurrence pattern.

We aim to extract the co-occurrence petterns mentioned above. This paper proposes a novel method of co-occurrence pattern extraction called *Co-occurring Cluster Mining*. Note that we do not consider the order of occurrence of AE events in the same basket. The reason is mentioned in section 3.2.

A conventional frequent pattern means an item set appearing frequently, whereas co-occurrence pattern means two sets composed of events that co-occur frequently. Therefore, the proposed method is regarded as a particular kind of clustering method considered the co-occurrence between clusters, rather than frequent pattern mining.

2.3 The Objective Function

In this section, the objective function is defined to search co-occurrence patterns. In searching, the most complex problem is to make clusters which satisfy

the correlation and similarity. We search the pairs of clusters $A, B \subset \mathcal{D}$ which maximize the following objective function:

$$L(A, B) = \{f(A, B)\}^{\alpha} \cdot \{g(A, B)\}^{(1-\alpha)}, \tag{1}$$

where the function $f(A, B)$ denotes the pattern correlation. The higher $f(A, B)$ value is, the more correlative pattern. For example, Jaccard coefficient and confidence as in association rule are used as $f(A, B)$. Jaccard coefficient is used in case of analyzing the ratio of the co-occurrence of event A and B. While confidence is used to the co-occurrence of event B under the situation that event A occurred. Note that because requirement 1 denotes the correlation among many separated baskets, the correlation in the short and sequential period must be excluded. Therefore, even if events A and B co-occur several times in the same basket, this is considered only once.

On the other hand, the function $g(A, B)$ denotes the pattern similarity. The higher $g(A, B)$ value is, the more similar clusters. For example, the distance between clusters, or the variance within each cluster are used as $g(A, B)$. Note that some definitions of the distance between clusters does not guarantee the monotonicity of cluster merge, e.g., centriod and median methods in hierarchical clustering. Therefore, we should avoid using those distances for $g(A, B)$. If only the distance between objects can be calculated, the average distance between objects is used as the variance in a cluster.

Since the co-occurrence patterns must satisfy the requirements of correlation and similarity at the same time, the objective function is defined as the product of $f(A, B)$ and $g(A, B)$. Generally speaking, the range of $f(A, B)$ is different from that of $g(A, B)$. Therefore, by normalizing as $f(A, B), g(A, B) \in [0, 1]$, both requirements of the correlation and similarity can be satisfied equally. The parameter $\alpha \in [0, 1]$ determines whether the correlation or similarity should be considered strongly. If α is close to 1, the similarity is considered more strongly than the correlation, and if α is close to 0, and vice versa. By maximizing the objective function by eq. (1), the co-occurrence patterns are obtained which satisfies the requirements of correlation and similarity. In addition, the requirement of frequency can be satisfied by extracting patterns which have higher support value than the pre-defined minimum support.

2.4 The Algorithm

The proposed method searches the pairs of clusters maximizing $L(A, B)$. The proposed method is based on an aggregative clustering, because of high computational complexity when using partition clustering like k-means. Partition clustering needs to be executed every time variables in the search are changed, in order to generate the candidate clusters of A and B. On the contrary, in aggregative clustering, once the merge process of clustering is obtained, co-occurrence patterns can be searched on the merge process. However, even in aggregative clustering, the pair of *Seeds*(starting point of clustering) is $O(N^2)$, and the expansion from each *Seed* will be $O(N)$. Therefore, the total computational complexity is $O(N^4)$. For the reduction of search space, we utilize a

dendrogram from the result of hierarchical clustering and search co-occurrence patterns on the obtained dendrogram. By using the dendrogram, although the degree of freedom about the decision of cluster shape decreases, the benefit is a great reduction of the search space to $O((N \log N)^2)$.

The algorithm of the proposed method is presented below. Here, assume a dendrogram by some hierarchical clustering has been obtained in advance.

Co-ocuurring cluster mining algorithm

Input: $L_{min}(A, B)$, $Supp_{min}(A, B)$, dendrogram by hierarchical clustering (HC), baskets of numerical event sequence $\mathcal{D} = \{Object_k\}_{k=1}^{N}$.
Output: Co-occurrence patterns $\{P(A, B)\}$.

BEGIN
  FOR i from 1 to N **DO**
    FOR $j(\neq i)$ from 1 to N **DO**
      Cluster $A \leftarrow Object_i$, Cluster $B \leftarrow Object_j$;
      Initialize $L_{Best}(A, B) \leftarrow 0$;
      WHILE (TRUE) **DO**
        IF $L(A, B) > L_{Best}(A, B)$ **THEN**
          $L_{Best}(A, B) \leftarrow L(A, B)$;
          Cluster $A_{Best} \leftarrow$Cluster A, Cluster $B_{Best} \leftarrow$Cluster B;
        END IF
        IF all pairs of clusters satisfying $A \cap B = \emptyset$ have been searched **THEN**
          BREAK;
        END IF
        Cluster $A \leftarrow Expand\_Cluster(A, HC)$;
        IF $A \cap B \neq \emptyset$ **THEN**
          Cluster $A \leftarrow Object_i$, Cluster $B \leftarrow Expand\_Cluster(B, HC)$;
        END IF
      END WHILE
      IF $L(A, B)_{Best} > L_{min}(A, B)$ and $Supp(A, B) > Supp_{min}(A, B)$ **THEN**
        Output $P(A, B) = \{A_{Best}, B_{Best}\}$;
      END IF
    END FOR
  END FOR
END

Here, Fig. 1 represents an example of the process of $Expand\_Cluster()$. $Expand\_Cluster()$ moves the current merge state from a certain node to the upper parent node, all leaf nodes (objects) which are children of this node belong to new cluster A. By the expansion of cluster A, label A is given to objects 1 and 6. And A and B co-occur in baskets 1 and 3, namely, a co-occurrence pattern (A, B) appears twice.

3 Application to AE Data

3.1 Damage Evaluation Test of Fuel Cells

A schematic diagram of the apparatus used to perform the SOFC damage test is shown in Fig. 2. The test section was initially heated up to 800°C in order to melt a soda glass ring and was then gradually decreased to room temperature. Note that this damage evaluation test was to rupture the cells intentionally while lowering the temperature. Therefore, the knowledge obtained through this

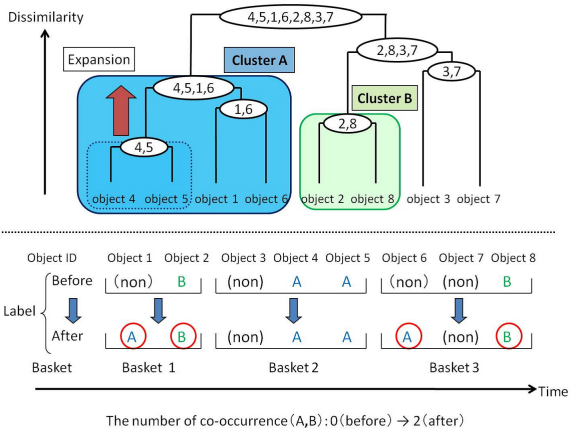


Fig. 1. The process of the proposed method. The above shows the dendrogram in the data space, and the below shows baskets in the time series.

experiment is not directly available to actual running the SOFC. However, it is sufficient to demonstrate and confirm the reasonableness of the proposed method. The AE measurement was performed using a wide-band piezoelectric transducer. The AE transducer was attached to an outer Al_2O_3 tube away from the heated section. The sampling rate is 1 MHz, and so the observable maximum frequency is 500 KHz. Running the SOFC for over 60 hours, 1,429 AE events were extracted using the burst extraction method [12,5].

Then, the same as the research by Fukui et al. [5], the AE events obtained from damage evaluation test are transformed into frequency spectrum data by Fast Fourier Transform (FFT). We obtained 1,429 frequency spectrum data each of which consists of 3,968 discrete points.

3.2 Division into Basket

Assume that the potential stress in a composite material is released after a large-energy AE event occurs, i.e., interactions of internal forces are reset. In this research, the observed AE event sequence was divided into baskets followed by the research of Kitagawa et al. [6], assuming a sequence until a large-energy AE event occurs to be a chain of damage progression. These baskets are used in the proposed method. Note that because the damage process of SOFC is a complicated system, it is difficult to extract co-occurrence patterns considering the order of occurrence or the time intervals between AE events exactly. Therefore, we do not consider the order of occurrence of AE events in the same basket, or the time intervals between AE events.

In this research, the energy threshold is $1,500\text{mV}^2$, which is also used in the research [6]. Then the AE event sequence is divided into 123 baskets. Fig. 3 shows an example of division into baskets.

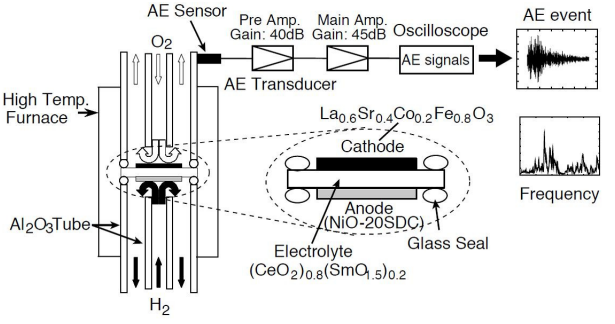


Fig. 2. SOFC damage test apparatus

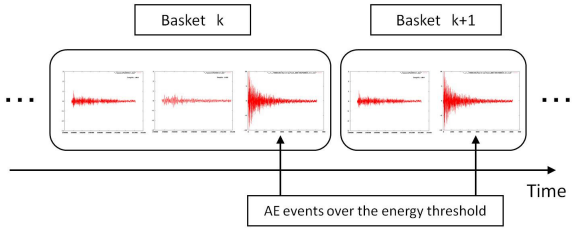


Fig. 3. An example of division into baskets (damage segments)

3.3 Calculation of Distance between AE Events

This work utilizes the result of the research by Fukui et al.[5] obtained by the kernel SOM. We regard the objects used for hierarchical clustering as the codebook vector (prototype). The self-organizing map[13] is an unsupervised neural network and a visualization technique by mapping the high dimensional feature space into the lower dimensional space (mainly two dimension). Kernel SOM[14] is the extended method of conventional SOM by improving the ability to express distribution of data with kernel method, which extends linear analysis method to non-linear method by mapping higher dimension. According to the earlier research, because the major damage types are already known on the kernel SOM, we can visually understand damage types with this result. Furthermore, the number of data points N is replaced to the number of prototypes $M (N >> M)$ by the quantization of the data space, the computational cost of searching co-occurring patterns is significantly reduced.

The distance between prototype vectors for the requirement of similarity (requirement 3) can be culculated as follows. Let M neurons of the prototype vectors be $\{\mathbf{m}_1, \cdots, \mathbf{m}_M\}$, where $\mathbf{m}_j = (m_{j,1}, \cdots, m_{j,v})$. In addition, let the position of M neurons in the topological layer be $\mathbf{r}_j = (\xi_j, \eta_j) : j = 1, \cdots, M$.

The number of neurons and the layout of the topological layer must be predefined, and a regular or hexagonal grid is normally used. Also, let a function $\phi : \Omega \to \mathcal{H}$ maps an original data space Ω to a high dimensional feature space $\mathcal{H}$. Then the prototype vector $\mathbf{m}_i$ is calculated by:

$$\mathbf{m}_i = \gamma_i \sum\nolimits_n h_{c(n),i} \phi(\mathbf{x}_n), \tag{2}$$

where $\gamma_i = 1/\sum_n h_{c(n),i}$ refers to the normalization factor. The neighborhood function $h_{i,j}$ is the Gaussian function: $h_{i,j} = \exp(-\|\mathbf{r}_i - \mathbf{r}_j\|^2/2\sigma^2)$, where σ refers to the radius which represents the influence range of neighborhood.

Although the mapping function $\phi(\mathbf{x}_n)$ cannot be calculated directlyC a kernel function can be defined as $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$, where $\langle , \rangle$ refers to the scalar product.

In this research, we use Kullback-Leibler (KL) kernel which was validated against the waveform data by Ishigaki et al.[15] and Fukui et al.[5].

Finally, the distance between prototype vectors $\mathbf{m}_i$ and $\mathbf{m}_j$ is culculated by:

$$\begin{aligned} d_{i,j} &= \|\mathbf{m}_i - \mathbf{m}_j\|^2 = \gamma_i^2 \sum_k \sum_l h_{c(k),i} h_{c(l),i} K(\mathbf{x}_k, \mathbf{x}_l) \\ &-2\gamma_i \gamma_j \sum_k \sum_l h_{c(k),i} h_{c(l),j} K(\mathbf{x}_k, \mathbf{x}_l) + \gamma_j^2 \sum_k \sum_l h_{c(k),j} h_{c(l),j} K(\mathbf{x}_k, \mathbf{x}_l). \end{aligned} \tag{3}$$

Since batch learning is used for kernel SOM, the neighborhood radius σ gradually decreases as learning is iterated. In culculating the distance of the codebook vector from the map obtained after the learning, we cannot decide the value of σ definitely. Therefore, the optimal neighborhood radius σ^* used for extraction of co-occurrence patterns is supposed to maximize the variance of the distances between the prototype vectors: $\sigma^* = \arg\max_\sigma V(d_{i,j})$. $\sigma^* = 0.1$ was linearly searched at intervals of 0.01. We use this $d_{i,j}$ as the distance between individual events.

3.4 The Design of the Object Function

This paper focuses on the co-occurrence relationship in certain damage period, namely about two sets of AE events A and B, we aim to extract damages co-occurring at the high probability. Therefore, we use Jaccard coefficient as $f(A, B)$:

$$f(A, B) = \frac{count(A \cap B)}{count(A \cup B)}, \tag{4}$$

where $count(A)$ is the number of baskets where event A appears.

Moreover, we cannot obtain the centroid of clusters but can obtain the distance between codebook vectors with d_{ave} as the average distance among all pairs of codebook vectors in the cluster. Hence, $g(A, B)$ is:

$$g(A, B) = 1 - \sqrt{d_{aveA} d_{aveB} / d_{aveALL}^2}, \tag{5}$$

Table 1. The average values of the objective function of the extracted 100 patterns in different hierarchical clustering methods; single linkage, complete linkage, group average, centroid, median, and Ward's method.

Method	Single	Complete	Average	Centroid	Median	Ward
Average	0.443	0.494	0.482	0.444	0.459	0.487

Table 2. The number of extracted damage patterns. The alphabets of damage types are listed in Table 3, and the inter-regions damage types are represented with ",".

Pattern	Number	Pattern	Number	Pattern	Number
(B)-(B)	2	(D)-(D)	1	(E)-(D),(E)	1
(B)-(C)	3	(D)-(E)	1	(F)-(A),(D)	1
(B)-(D)	2	(E)-(E)	4	(F)-(D),(E)	1
(B)-(E)	2	(E)-(F)	5	(A),(D)-(D),(E)	1
(C)-(C)	1	(E)-(A),(D)	3	(D),(E)-(D),(E)	1

where d_{ave} is normalized divided by d_{aveALL} of the largest cluster so that $g(A, B) \in [0, 1]$. To consider the corrrelation and similarity of patterns as equally as possible, the parameter α in eq. (1) is set to 0.5.

3.5 The Results of Extracted Damage Patterns

The topology of kernel SOM is two dimensional square grid, and the number of neurons is 15×15.

First, Table 1 shows the average values of the objective function in different hierarchical clustering methods. The values are averaged by the extracted 100 patterns when the minimum support is 0.04. The complete linkage method shows the best result. The following all results are obtained by using the complete linkage method in the hierarchical clustering.

Next, the representative extracted damage patterns are explained. Two experts of SOFC of the co-authors interpreted damage patterns. Table 2 shows the estimated interpretation of extracted damage patterns. As the parameters of pattern extraction, the minimum object function is 0.47, the minimum support is 0.04, 29 patterns were extracted. The computational time when using 1429 individual objects was 888.7 (sec) with Intel Xeon CPU 2.66GHz and 6GB RAM. While, when using prototypes of the kernel SOM, the computational time in 225 objects was 25.6 (sec).

In addition, Fig. 4 shows an example of the result of extracted damage pattern on the result of kernel SOM. The correspondence of the regions on the map to damage types is shown in Table 3. This damage types and frequent period are already known by the research of Fukui et al. [5]. Each damage pattern is

Table 3. The major damage types corresponding to the map in Fig. 4

Region	Damage type
(A)	squeaking of the members during heating
(B)	progression of the initial cracks
(C)	squeaking of the members followed by (B)
(D)	cracks in the electrolyte
(E)	cracks in the glass seal
(F)	cracks in and exfoliation of the electrode

distinguished by using different colors, and the typical waveforms and spectra of the damage type are shown.

Valid results based on the knowledge of SOFC experts: Damage pattern 1 is a co-occurring pattern of (B) the progression of the initial cracks and (D) cracks in the electrolyte. In pattern 1, AE events on the top of map are a part of (B) of which these AE events occur in the latter period. Therefore, damage pattern 1 is interpreted that the progression of the initial cracks causes cracks in the electrolyte. According to table 2, we can know that progression of the initial cracks co-occurs with various damages. So we estimate that progression of the initial cracks is the starting points of various damages.

Next, damage pattern 2 is the co-occurring patterns of the cracks in the glass seal and cracks in and exfoliation of the electrode. Especially, the damage type which influences cracks in and exfoliation of the electrode are cracks in the glass seal which occurs in the latter period of cracks of the glass seal. The glass seal changes its state by the temperature, and in the period mentioned above, the temperature is decreasing and glass seal is congealed at the temperature of damage pattern 2. The glass seal and the electrode are not connected directly, but it is supposed that the shrinking and transformation of the cell due to the congelation of the glass seal produces the indirect mechanical effect.

Novel results: According to Table 2, no damage patterns which include both regions (D) and (F) are extracted. In spite of the fact that the electrolyte and the electrode are connected, damage patterns which include them were not extracted at all. This result was interesting to SOFC experts.

Next, since damage pattern 3 exists in the inter-regions, damage pattern 3 may contain novel damage types. Since these damages cause AE events contain high peaks in the low frequency of the spectrum, the damages between regions (A) and (D) are estimated as the exfoliation of the electrolyte, and the damages between regions (D) and (E) are estimated as the exfoliation of the electrolyte or the glass seal. Damages of pattern 3 have never discovered from the earlier research based only on the occurrence frequency of each AE event. Taking the co-occurrence relationship of AE events into consideration, damage pattern 3 is discovered for the first time.

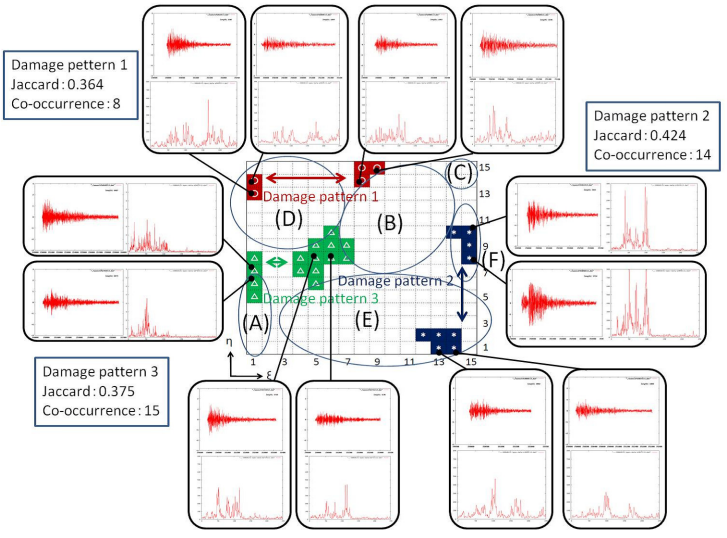


Fig. 4. An example of extracted damage patterns. The central map is the classification result by the kernel SOM.

4 Conclusion

In this paper, we proposed the novel extraction method of co-occurrence patterns against numerical data: *Co-occurring Cluster Mining*. The proposed method determines the area (or the components) of two co-occurring clusters considering co-occurrence between clusters and simultaneously similarity in each cluster. We applied the proposed method for AE events obtained from damage evaluation test of SOFC. As a result, damage patterns which demonstrate major mechanical correlations of SOFC were extracted, including unexpected but valuable damage patterns.

Furthermore, we will apply the proposed method for various numerical data such as earthquake wave or the track of point on dynamic image, and demonstrate the general-purpose of the proposed method.

Acknowledgements. This work was supported in part by the Management Expenses Grants for National Universities Corporations from the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT) and also KAKENHI (21700165).

References

1. Krishnamurthy, R., Sheldon, B.W.: Stress due to oxygenpotential gradientsinnon-stoichiometricoxides. Journal of Acta Materialia 52, 1807–1822 (2004)

2. Sato, K., Omura, H., Hashida, T., Yashiro, K., Kawada, T., Mizusaki, J., Yugami, H.: Tracking the onset of damage mechanism in ceria-based solid oxide fuel cells under simulated operating conditions. Journal of Testing and Evaluation 34(3), 246–250 (2006)
3. Rippengill, S., Worden, K., Holford, K.M., Pullin, R.: Automatic classification of acoustic emission patterns. Journal for Experimental Mechanics: Strain 39(1), 31–41 (2003)
4. Godin, N., Huguet, S., Gaertner, R.: Influence of hydrolytic ageing on the acoustic emission signatures of damage mechanisms occurring during tensile tests on a polyester composite: Application of a Kohonen's map. Composite Structures 72(1), 79–85 (2006)
5. Fukui, K., Akasaki, S., Sato, K., Mizusaki, J., Moriyama, K., Kurihara, S., Numao, M.: Visualization of Damage Progress in Solid Oxide Fuel Cells. Journal of Environment and Engineering 6(3), 499–511 (2011)
6. Kitagawa, T., Fukui, K.-i., Sato, K., Mizusaki, J., Numao, M.: Extraction of Essential Events with Application to Damage Evaluation on Fuel Cells. In: Hatzilygeroudis, I., Prentzas, J. (eds.) Combinations of Intelligent Methods and Applications. SIST, vol. 8, pp. 89–108. Springer, Heidelberg (2011)
7. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proc. of 20th International Conference on Very Large Databases (ICVLD), pp. 487–499 (1994)
8. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: Proc. of the 1998 ACM SIGMOD International Conference on Management of Data (ICMD), pp. 94–105 (1998)
9. Mitsunaga, Y., Washio, T., Motoda, H.: Mining Quantitative Frequent Itemsets Using Adaptive Density-Based Subspace Clustering. In: Proc. of the 5th International Conference on Data Mining (ICDM), pp. 793–796 (2005)
10. Honda, R., Konishi, O.: Temporal Rule Discovery for Time-Series Satellite Images and Integration with RDB. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, pp. 204–215. Springer, Heidelberg (2001)
11. Yairi, T., Ishihama, N., Kato, Y., Hori, K., Nakasuka, S.: Anomaly Detection Method For Spacecrafts Based on Association Rule Mining. Journal of Space Technology and Science 17(1), 1–10 (2001)
12. Kleinberg, J.: Bursty and hierarchical structure in streams. In: Proc. the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2002), pp. 91–101 (2002)
13. Kohonen, T.: Self-organizing maps. Springer (1995)
14. Boulet, R., Jouve, B., Rossi, F., Villa, N.: Batch Kernel SOM and Related Laplacian Method for Social Network Analysis. Neurocomputing 71, 1257–1273 (2008)
15. Ishigaki, T., Higuchi, T.: Dynamic Spectrum Classification by Kernel Classifiers with Divergence-Based Kernels and its Applications to Acoustic Signals. International Journal of Knowledge Engineering and Soft Data Paradigms 1(2), 173–192 (2009)

New Exact Concise Representation of Rare Correlated Patterns: Application to Intrusion Detection

Souad Bouasker[1], Tarek Hamrouni[1], and Sadok Ben Yahia[1,2]

[1] LIPAH, Computer Science Department, Faculty of Sciences of Tunis, Tunis, Tunisia
[2] Institut TELECOM, TELECOM SudParis, UMR 5157 CNRS SAMOVAR, France

Abstract. During the last years, many works focused on the exploitation of rare patterns. In fact, these patterns allow conveying knowledge on unexpected events. Nevertheless, a main problem is related to their very high number and to the low quality of several mined rare patterns. In order to overcome these limits, we propose to integrate the correlation measure *bond* aiming at only mining the set of rare *correlated* patterns. A characterization of the resulting set is then detailed, based on the study of constraints of different natures induced by the rarity and the correlation. In addition, based on the equivalence classes associated to a closure operator dedicated to the *bond* measure, we propose a new exact concise representation of rare correlated patterns. We then design the new RCPRMINER algorithm allowing an efficient extraction of the proposed representation. The carried out experimental studies prove the compactness rate offered by our approach. We also design an association rules based classifier and we prove its effectiveness in the context of intrusion detection.

Keywords: Concise representation, Constraint, Rarity, Correlation, Closure operator, Equivalence class.

1 Introduction and Motivations

Recently, rare pattern mining has been proved to be of actual added value in many application fields such as the intrusion detection, the analysis of criminal data, the pharmacovigilance, etc. [7]. In fact, rare patterns can identify unexpected events or exceptions [15], since they have a very low frequency in the data base. In practice, the exploitation of rare patterns is hampered by the high number and the low quality of the extracted rare patterns. Thus, an extracted rare pattern may not represent any useful information whenever it is composed only by items among which there is no *semantic* link. In this situation, integrating correlation measures would be of benefit by only mining *Rare correlated patterns*. These latter patterns offer a strong semantic link among their items. Indeed, an interesting rare pattern is that which appears a very small number of times in the database but has items that are strongly linked w.r.t. a correlation metric.

In this paper, we focus on the extraction of an exact concise representation of rare correlated patterns w.r.t. to the *bond* correlation measure [10]. This measure is redefined in this work as the ratio between the conjunctive support of a pattern and its disjunctive support. Indeed, although used in many works under various names like *extended Jaccard measure*, *coherence*, *Tanimoto coefficient*, the link between the expression of this

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 61–72, 2012.

measure and the disjunctive support in the denominator part has never been established in the literature. Our choice of this measure is motivated by the theoretical framework presented in [10,14], in addition to the structural study that was done in [12]. Furthermore, it has been proved in [14] that the *bond* measure fulfills the theoretical properties that any measure of quality dedicated to rare association rule should have. Moreover, the authors in [12] proposed a generic approach for correlated patterns mining based on the *bond* measure. Note, however, that the study of rare correlated patterns was not previously carried out in the literature.

From the computational point of view, the integration of the *bond* measure within the mining process of rare patterns is a very challenging task. Indeed, the correlated patterns associated to the *bond* measure verify an anti-monotone constraint and then induce an *order ideal* [5] in the pattern lattice. In opposition to this, rare patterns form an *order filter* [5] in the pattern lattice since they fulfill a monotone constraint. Therefore, the set of rare correlated patterns result from the intersection of two theories [9] induced by the constraints of correlation and rarity. The set of rare correlated patterns is then more complicated to be mined than any set of patterns induced by one or more constraints of the same nature [4]. We thus provide in this paper a thorough characterization of this set of patterns based on the notion of equivalence class. In our case, equivalence classes are induced by the closure operator associated to the *bond* measure.

To the best of our knowledge, there is no previous study in the literature that has been dedicated to the extraction of a concise representation of patterns fulfilling both the rarity and the correlation constraints. Worth of mention that the new proposed approach is generic and can then be applied to any set of rare correlated pattern according to any correlation measure which shares the same structural properties as the *bond* measure, *e.g.*, the *all-confidence* measure [10] [1]

The remainder of the paper is organized as follows: Section 2 presents basic notions used throughout this work. In Section 3, we characterize the set of all rare correlated patterns by studying the associated constraints. We also introduce the associated new exact concise representation. Section 4 is dedicated to the description of the RCPRMINER algorithm allowing the extraction of the proposed representation. The empirical studies are provided in Section 5. Section 6 illustrates the application of our approach in intrusion detection. The conclusion and perspectives are sketched in Section 7.

2 Basic Notions

We start by presenting the key notions related to our work. We first define a dataset.

Definition 1. (***Dataset***) *A dataset is a triplet* $\mathcal{D} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$ *where* $\mathcal{T}$ *and* $\mathcal{I}$ *are, respectively, a finite set of transactions and items, and* $\mathcal{R} \subseteq \mathcal{T} \times \mathcal{I}$ *is a binary relation between the transaction set and the item set. A couple* $(t, i) \in \mathcal{R}$ *denotes that the transaction* $t \in \mathcal{T}$ *contains the item* $i \in \mathcal{I}$.

In this work, we are mainly interested in itemsets as a class of patterns. The two main kinds of support a pattern can have are defined as follows, for any non-empty pattern I:
- **Conjunctive support**: $Supp(\wedge I) = \mid \{t \in \mathcal{T} \mid (\forall\, i \in I, (t, i) \in \mathcal{R})\} \mid$
- **Disjunctive support**: $Supp(\vee I) = \mid \{t \in \mathcal{T} \mid (\exists\, i \in I, (t, i) \in \mathcal{R})\} \mid$

[1] Mathematically equivalent to the *h-confidence* measure [16].

Table 1. An example of a dataset

	A	B	C	D	E
1	×		×	×	
2		×	×		×
3	×	×	×		×
4		×			×
5	×	×	×		×

Example 1. Let us consider the dataset given by Table 1. We have $Supp(\wedge \mathtt{AD}) = |\{1\}| = 1$ and $Supp(\vee \mathtt{AD}) = |\{1, 3, 5\}| = 3$.[2]

We distinguish, given a minimum support threshold [1], between *frequent* and *rare* patterns. These latter patterns are defined as follows.

Definition 2. (***Rare patterns***) *The set of rare patterns is defined by:* $\mathcal{RP} = \{I \subseteq \mathcal{I} | Supp(\wedge I) < minsupp\}$.

Among the elements of $\mathcal{RP}$, we distinguish the smallest rare patterns according to set-inclusion relation. These patterns constitute the set $\mathcal{MIN}\mathcal{RP}$ defined as follows:

Definition 3. (***Minimal rare patterns***) *The* $\mathcal{MIN}\mathcal{RP}$ *set of minimal rare pattern is composed by rare patterns having no rare proper subset. It is equal to:* $\mathcal{MIN}\mathcal{RP} = \{I \in \mathcal{RP} | \forall I_1 \subset I: I_1 \notin \mathcal{RP}\}$.

Example 2. Let us consider the dataset given by Table 1 for *minsupp* = 4. We have, for example, the pattern $\mathtt{BC} \in \mathcal{RP}$ since $Supp(\wedge \mathtt{BC}) = 3 < 4$. We also have the pattern $\mathtt{BC} \in \mathcal{MIN}\mathcal{RP}$ since $Supp(\wedge \mathtt{BC}) = 3 < 4$ and, on the other hand, $Supp(\wedge \mathtt{B}) = Supp(\wedge \mathtt{C}) = 4$. In this case, $\mathcal{MIN}\mathcal{RP} = \{\mathtt{A}, \mathtt{D}, \mathtt{BC}, \mathtt{CE}\}$.

In the following, we define *monotone* and *anti-monotone* constraints [4,11].

Definition 4. (***Monotone/Anti-Monotone constraint***) *Let* Q *be a constraint,*

- Q *is anti-monotone if* $\forall I \subseteq \mathcal{I}$, $\forall I_1 \subseteq I$: I *fulfills* $Q \Rightarrow I_1$ *fulfills* Q
- Q *is monotone if* $\forall I \subseteq \mathcal{I}$, $\forall I_1 \supseteq I$: I *fulfills* $Q \Rightarrow I_1$ *fulfills* Q

The constraint of rarity is a monotone constraint, *i.e.*, $\forall I, I_1 \subseteq \mathcal{I}$, if $I_1 \supseteq I$ and $Supp(\wedge I) < minsupp$, then $Supp(\wedge I_1) < minsupp$ since $Supp(\wedge I_1) \leq Supp(\wedge I)$. Thus, it induces an *order filter* [5] on the set of all the subsets of $\mathcal{I}$, $\mathcal{P}(\mathcal{I})$. Worth of mention that the frequency constraint induces an *order ideal* [5].

Definition 6 presents the set of correlated patterns according to the *bond* measure [10] which is redefined here as given in Definition 5.

Definition 5. (***The bond measure***) *The bond measure of a non-empty pattern* $I \subseteq \mathcal{I}$ *is defined as follows:*

$$bond(I) = \frac{Supp(\wedge I)}{Supp(\vee I)}$$

Definition 6. (***Correlated patterns***) *Considering a minimum correlation threshold minbond, the set* $\mathcal{CP}$ *of correlated patterns is equal to:* $\mathcal{CP} = \{I \subseteq \mathcal{I} \mid bond(I) \geq minbond\}$.

[2] We use a separator-free form for the sets, *e.g.*, AD stands for the set of items {A, D}.

The constraint of correlation is an anti-monotone constraint, *i.e.*, $\forall\ I,\ I_1 \subseteq \mathcal{I}$, if $I_1 \subseteq I$, then $bond(I_1) \geq bond(I)$. Therefore, the set $\mathcal{CP}$ of correlated patterns forms an *order ideal* [5] on $\mathcal{P}(\mathcal{I})$.

In the following, we will need the set composed by the maximal correlated patterns which constitute the positive border of correlated patterns. This set is defined as follows:

Definition 7. (***Maximal correlated patterns***) *The set of maximal correlated patterns, denoted $\mathcal{M}ax\mathcal{CP}$, is composed by correlated patterns having no correlated proper superset, i.e., $\mathcal{M}ax\mathcal{CP} = \{I \in \mathcal{CP} | \forall\ I_1 \supset I: I_1 \notin \mathcal{CP}\}$.*

Example 3. Consider the dataset illustrated by Table 1. For *minbond* = 0.2, we have $bond(\texttt{BCE}) = \frac{3}{5} = 0.6 \geq 0.2$. Therefore, the pattern BCE is a correlated one. In addition, whatever the strict superset of BCE, this superset is not correlated. In this case, we have $\mathcal{M}ax\mathcal{CP} = \{\texttt{ACD}, \texttt{ABCE}\}$.

Now we focus on the closure operator associated to the *bond* measure.

Definition 8. (***The operator*** f_{bond}) *The closure operator f_{bond}: $\mathcal{P}(\mathcal{I}) \rightarrow \mathcal{P}(\mathcal{I})$ associated to the bond measure is defined as follows: $f_{bond}(I) = I \cup \{i \in \mathcal{I} \setminus I \mid bond(I) = bond(I \cup \{i\})\}$.*

The closure of a pattern I by f_{bond}, *i.e.* $f_{bond}(I)$, corresponds to the maximal set of items containing I and sharing the same *bond* value with I. It can be easily proven that $f_{bond}(I)$ is equal to the intersection of its conjunctive closure f_c and its disjunctive one f_d. Indeed, according to the framework proposed in [13], *bond* measure is a condensable function based on both preserving functions, namely the conjunctive support and the disjunctive support.

Example 4. Consider the dataset illustrated by Table 1. For *minbond* = 0.2, we have $f_{bond}(\texttt{AB}) = \texttt{ABCE}$ since C and E preserve the *bond* value of AB.

We focus now on the equivalence classes induced by the f_{bond} closure operator.

Definition 9. (***Equivalence class associated to the closure operator*** f_{bond}) *An equivalence class associated to the closure operator f_{bond} is composed by all the patterns having the same closure by the operator f_{bond}. Let $[I]$ be the equivalence class to which belongs the pattern I. $[I]$ is formally defined as follows: $[I] = \{I_1 |\ f_{bond}(I) = f_{bond}(I_1)\}$.*

In each class, all the patterns share the same *bond* value as well as the same conjunctive, disjunctive, and negative supports. Therefore, all the patterns belonging to the same class, induced by f_{bond}, appear exactly in the same transactions. Besides, these patterns characterize the same set of transactions. Indeed, each transaction necessarily contains a non-empty subset of each pattern of the class.

In the next section, we will carry out a detailed study of the rare correlated patterns.

3 Characterization of the Rare Correlated Patterns

3.1 Definition and Properties

The set of rare correlated patterns is defined as follows:

Definition 10. (***Rare correlated patterns***) *Considering the support threshold minsupp and the correlation threshold minbond, the set of rare correlated patterns, denoted $\mathcal{RCP}$, is equal to:* $\mathcal{RCP} = \{I \subseteq \mathcal{I} \mid Supp(\wedge I) < minsupp \text{ and } bond(I) \geq minbond\}$.

Example 5. Consider the dataset illustrated by Table 1. For *minsupp* = 4 and *minbond* = 0.2, the set $\mathcal{RCP}$ consists of the following patterns where each triplet represents the pattern, its conjunctive support value and its *bond* value: $\mathcal{RCP}$ = {(A, 3, $\frac{3}{3}$), (D, 1, $\frac{1}{1}$), (AB, 2, $\frac{2}{5}$), (AC, 3, $\frac{3}{4}$), (AD, 1, $\frac{1}{3}$), (AE, 2, $\frac{2}{5}$), (BC, 3, $\frac{3}{5}$), (CD, 1, $\frac{1}{4}$), (CE, 3, $\frac{3}{5}$), (ABC, 2, $\frac{2}{5}$), (ABE, 2, $\frac{2}{5}$), (ACD, 1, $\frac{1}{4}$), (ACE, 2, $\frac{2}{5}$), (BCE, 3, $\frac{3}{5}$), (ABCE, 2, $\frac{2}{5}$)}. This set is depicted by Figure 1. The support shown at the top left of each frame represents the conjunctive support. As shown in Figure 1, the rare correlated patterns are then localized below the border shown in red of the anti-monotone constraint of correlation, composed by the elements of $\mathcal{M}ax\mathcal{CP}$, and over the border shown in black of the monotone constraint of rarity, composed by the elements of $\mathcal{M}in\mathcal{RP}$.

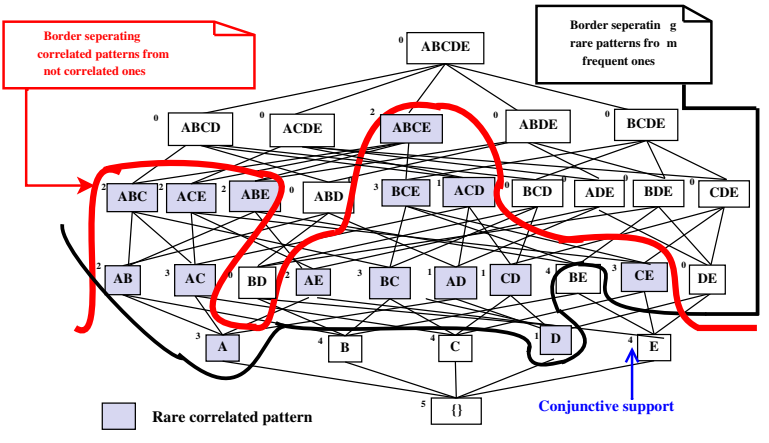


Fig. 1. Localization of the rare correlated patterns for *minsupp* = 4 and *minbond* = 0.2

Therefore, the localization of the rare correlated patterns is much more complex than the localization of theories corresponding to constraints of the same nature.

Interestingly enough, the use of *bond* allows improving the quality of mined patterns by only retaining those containing strongly correlated items. However, the number of these patterns is not necessarily reduced which may hamper its practical use. To losslessly reduce such amount of information, we propose to define a new exact concise representation of the $\mathcal{RCP}$ set. An exact representation of rare correlated patterns should determine, for an arbitrary pattern, whether it is rare correlated or not. If it is a rare correlated one, then this representation must allow faithfully deriving the values of its support and its *bond* measure. In this respect, the proposed representation in this work will be shown to be perfect in the sense that its size never exceeds that of the whole set of rare correlated patterns. It also allows a better exploitation and management of the

extracted knowledge. In addition, since the representation, that we introduce, is lossless, it allows to derive, whenever of need, the whole set of rare correlated patterns.

The new exact concise representation of rare correlated patterns is based on the notion of equivalence class. Equivalence classes allow us to only keep track of non-redundant patterns. Indeed, we retain for each class only the maximal and the minimal ones. The next subsection details our approach.

3.2 Characterization of the Rare Correlated Equivalence Classes

Based on Definition 9, the elements of the same equivalence class have the same behavior w.r.t. both the correlation and the rarity constraints. In fact, for a correlated equivalence class, *i.e.* a class which contains correlated patterns, all of them could be rare or frequent. The application of f_{bond} then provides a more selective process to only extract representative rare correlated patterns of each class. The $\mathcal{RCP}$ set of rare correlated patterns is then split into disjoint equivalence classes – the rare correlated equivalence classes – in which the closed pattern is the largest one w.r.t. the set-inclusion relation. The smallest, w.r.t. the set-inclusion relation, patterns in a class are the minimal rare correlated patterns. The set of these particular patterns are formally defined as follows:

Definition 11. (***Closed rare correlated patterns***) *The $\mathcal{CRCP}$ set of closed rare correlated patterns is equal to:* $\mathcal{CRCP} = \{I \in \mathcal{RCP} | \forall\ I_1 \supset I$: $bond(I) > bond(I_1)\}$.

Definition 12. (***Minimal rare correlated patterns***) *The $\mathcal{MRCP}$ set of minimal rare correlated patterns is equal to:* $\mathcal{MRCP} = \{I \in \mathcal{RCP} | \forall\ I_1 \subset I$: $bond(I) < bond(I_1)\}$.

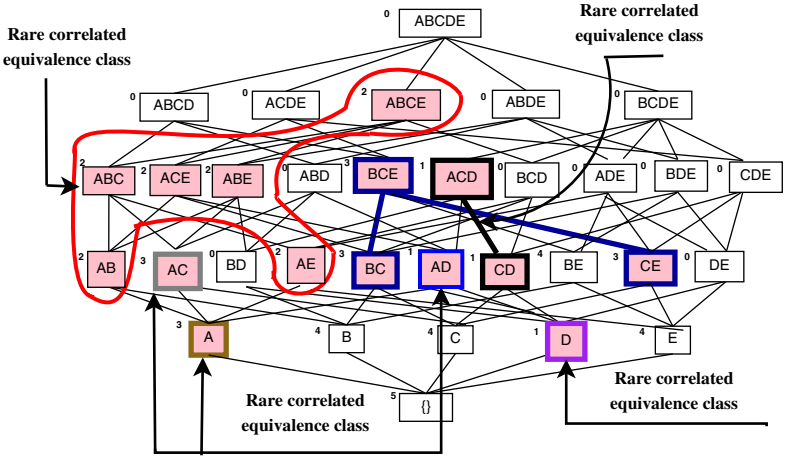


Fig. 2. An example of rare correlated equivalence classes for *minsupp* = 4 and *minbond* = 0.2

Example 6. Consider the dataset $\mathcal{D}$ illustrated by Table 1. For *minsupp* = 4 and *minbond* = 0.2, Figure 2 shows the rare correlated equivalence classes. We have, the set $\mathcal{CRCP}$ = {A, D, AC, AD, ACD, BCE and ABCE}. Whereas, the set $\mathcal{MRCP}$ is equal to: $\mathcal{MRCP}$ = {A, D, AB, AC, AD, AE, BC, CD and CE}. As shown by Figure 2, the patterns A, D, AC and AD are closed and at the same time minimal. Their equivalence classes then contain a unique element.

Before introducing our new concise representation, it is worth mentioning that the notions of closed patterns and minimal generators were also simultaneously used in [8] in order to offer a lossless concise representation of frequent itemsets. Now, based on the two previous sets, we define our new exact concise representation $\mathcal{RCPR}$.[3]

Definition 13. (***The*** $\mathcal{RCPR}$ ***representation***) *The* $\mathcal{RCPR}$ *representation is equal to:* $\mathcal{RCPR} = \mathcal{CRCP} \cup \mathcal{MRCP}$.

Example 7. Consider the dataset illustrated by Table 1 for *minsupp* = 4 and *minbond* = 0.2. According to the previous example, we have the $\mathcal{RCPR}$ representation composed by: (A, 3, $\frac{3}{3}$), (D, 1, $\frac{1}{1}$), (AB, 2, $\frac{2}{5}$), (AC, 3, $\frac{3}{4}$), (AD, 1, $\frac{1}{3}$), (AE, 2, $\frac{2}{5}$), (BC, 3, $\frac{3}{5}$), (CD, 1, $\frac{1}{4}$), (CE, 3, $\frac{3}{5}$), (ACD, 1, $\frac{1}{4}$), (BCE, 3, $\frac{3}{5}$) and (ABCE, 2, $\frac{2}{5}$).

The following theorem proves that the $\mathcal{RCPR}$ representation is a lossless concise representation of the $\mathcal{RCP}$ set. In this respect, both sets $\mathcal{MRCP}$ and $\mathcal{CRCP}$ composing $\mathcal{RCPR}$ are required for the exact regeneration of the set $\mathcal{RCP}$. The elements of the former set are indeed required for ensuring the rarity property of an arbitrary pattern. While the latter set is used for checking the correlation property and for exactly deriving its *bond* and support values.

Theorem 1. *The* $\mathcal{RCPR}$ *representation is an exact concise representation of the* $\mathcal{RCP}$ *set of rare correlated patterns.*

Proof. Let $I \subseteq \mathcal{I}$. We distinguish between three different cases:

a) If $I \in \mathcal{RCPR}$, then I is a rare correlated pattern and we have its support and its *bond* values in the representation.

b) If $\nexists\, J \in \mathcal{RCPR}$ such that $J \subseteq I$ or $\nexists\, Z \in \mathcal{RCPR}$ such that $I \subseteq Z$, then $I \notin \mathcal{RCP}$ since I does not belong to any rare correlated equivalence class.

c) $I \in \mathcal{RCP}$. Indeed, J and Z exist (otherwise I fulfills the conditions of the case b). Thus, I is correlated since it is included in a correlated pattern, namely Z. It is also rare since it contains a rare pattern, namely J. In this case, it is sufficient to localize the f_{bond} closure of I, say F. The closed pattern F belongs to $\mathcal{RCPR}$ since I is rare correlated and $\mathcal{RCPR}$ includes the $\mathcal{CRCP}$ set of closed rare correlated patterns. Therefore, $F = min_{\subseteq}\{I_1 \in \mathcal{RCPR} |\ I \subseteq I_1\}$. Since f_{bond} preserves the *bond* value and the conjunctive support, we then have: $bond(I) = bond(F)$ and $Supp(\wedge I) = Supp(\wedge F)$. ♢

Example 8. Consider the $\mathcal{RCPR}$ representation illustrated by Example 7. Let us consider each case separately. The pattern AD $\in \mathcal{RCPR}$. Thus, we have its support equal to 1 and its *bond* value equal to $\frac{1}{3}$. Although the pattern BE is included in two patterns from the $\mathcal{RCPR}$ representation, namely BCE and ABCE, BE $\notin \mathcal{RCP}$ since no element of $\mathcal{RCPR}$ is included in BE. Consider now the pattern ABC. There are two patterns of $\mathcal{RCPR}$ which allow determining that the pattern ABC is a rare correlated one, namely AB and ABCE, since AB $\subseteq$ ABC $\subseteq$ ABCE. The smallest pattern in $\mathcal{RCPR}$ which cover ABC, *i.e.* its closure, is ABCE. Then, $bond$(ABC) = $bond$(ABCE) = $\frac{2}{5}$, and $Supp(\wedge$ABC$)$ = $Supp(\wedge$ABCE$)$ = 2.

[3] $\mathcal{RCPR}$ stands for **R**are **C**orrelated **P**attern **R**epresentation.

The proof of Theorem 1 clearly highlights that it is straightforward the way queries over of the proposed representation would be carried out w.r.t. a given arbitrary pattern as well as the derivation of the whole set of rare correlated patterns. It is also important to mention that the $\mathcal{RCPR}$ representation is a *perfect cover* of the $\mathcal{RCP}$ set, *i.e.*, the size of $\mathcal{RCPR}$ *never exceeds* that of the $\mathcal{RCP}$ set whatever the dataset and the used *minsupp* and *minbond* values. It is in fact always true that $(\mathcal{CRCP} \cup \mathcal{MRCP}) \subseteq \mathcal{RCP}$.

In the following, we introduce the RCPRMINER algorithm dedicated to the extraction of the $\mathcal{RCPR}$ representation.

4 The RCPRMINER Algorithm

The pseudo-code of RCPRMINER is shown by Algorithm 1. RCPRMINER is a levelwise algorithm which takes as an input a dataset $\mathcal{D}$, a minimum support threshold *minsupp* and a minimum correlation threshold *minbond*. This algorithm allows the determination of the $\mathcal{MRCP}$ and the $\mathcal{CRCP}$ sets which constitute the $\mathcal{RCPR}$ representation.

Algorithm 1. RCPRMINER

Data: A dataset $\mathcal{D} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$, *minbond*, and *minsupp*.
Results: The exact concise representation $\mathcal{RCPR} = \mathcal{MRCP} \cup \mathcal{CRCP}$.
1 **Begin**
2 $\mathcal{RCPR} := \emptyset$; $\mathcal{C}and_0 := \{\emptyset\}$;
3 /* **The first step** */
4 $\mathcal{M}ax\mathcal{CP}$:= MAXCP_EXTRACTION($\mathcal{D}$, *minbond*);
5 /* **The second step** */
6 $\mathcal{M}ax\mathcal{CFP} := \{X \in \mathcal{M}ax\mathcal{CP} \mid X.ConjS \geq minsupp\}$ /* $X.ConjS$ denotes the conjunctive support of X */;
7 $\mathcal{M}ax\mathcal{RCP} := \{X \in \mathcal{M}ax\mathcal{CP} \mid X.ConjS < minsupp\}$;
8 $\mathcal{PC}and_1 := \{i \mid i \in \mathcal{I}\}$ /* $\mathcal{PC}and_n$ stands for Potential Candidates of size n */;
9 **While** ($\mathcal{PC}and_n \neq \emptyset$) **Do**
10 /* *Pruning of potential candidate patterns* */
11 $\mathcal{C}and_n := \mathcal{PC}and_n \setminus \{X_n \in \mathcal{PC}and_n \mid (\exists\, Z \in \mathcal{M}ax\mathcal{CFP}: X_n \subseteq Z)$ or $(\nexists\, Z \in \mathcal{M}ax\mathcal{RCP}: X_n \subseteq Z)$ or $(\exists\, Y_{n-1} \subset X_n: Y_{n-1} \notin \mathcal{C}and_{n-1})\}$;
12 /* *Determination of the minimal rare correlated patterns of size n and computation of their closures* */
13 $\mathcal{RCPR} := \mathcal{RCPR} \cup$ MRCP_CRCP_COMPUTATION($\mathcal{D}, \mathcal{C}and_n$, *minsupp*);
14 $n := n + 1$;
15 $\mathcal{PC}and_n$:= APRIORI-GEN($\mathcal{C}and_{n-1}$) ;
16 **Return** $\mathcal{RCPR}$;
17 **End**

The RCPRMINER algorithm mainly operates in two steps. The first step consists in extracting the maximal correlated patterns from the extraction context through the invocation of the dedicated MAXCP_EXTRACTION procedure (*cf.* Line 4). The second step consists in integrating the constraint of rarity and the obtained maximal correlated patterns in a mining process of $\mathcal{RCPR}$. In this situation, the set $\mathcal{PC}and_n$ of potential candidates of size n is obtained using the classical APRIORI-GEN procedure (*cf.* Line 15) from the retained candidates of size (n - 1). Once obtained, the set elements

of $\mathcal{PC}and_n$ are pruned (*cf.* Line 11) using several pruning strategies to yield the set $\mathcal{C}and_n$. The used pruning strategies are as follows:

(***i***) **The pruning of the candidates which are included in a maximal correlated frequent pattern.**

(***ii***) **The pruning of the candidates which are not included in a maximal rare correlated pattern.**

(***iii***) **The pruning based on the order ideal of the minimal correlated patterns**.

Recall that the set of minimal correlated patterns induces an order ideal property. Therefore, every minimal correlated candidate, having a non minimal correlated subset, will be pruned since it will not be a minimal correlated pattern. In this respect, within the MRCP_CRCP_COMPUTATION procedure (*cf.* Line 13), whose pseudo-code is omitted here for lack of available space, the minimal rare correlated patterns are determined among the retained candidates in $\mathcal{C}and_n$ (*cf.* Line 11). This is done by comparing their *bond* values to those of their respective immediate subsets. Then, minimal rare correlated patterns are inserted into the $\mathcal{MRCP}$ set. Their closures are after that computed according to the f_{bond} closure operator, and then, inserted in the $\mathcal{CRCP}$ set. From the computational point of view, it is important to mention that the localization of the border composed by the maximal correlated patterns is an NP-hard problem [3]. Therefore, this task constitutes the most consuming part, w.r.t. execution time, in RCPRMINER.

The next section experimentally studies the $\mathcal{RCPR}$ representation compactness.

5 Experimental Results

In this section, our main objective is to show, through extensive experiments, that the $\mathcal{RCPR}$ representation provides interesting compactness rates compared to the whole set of rare correlated patterns. All experiments were carried out on a PC equipped with a 2.7 GHz Intel Dual Core processor $E5400$ and 4 GB of main memory, running the Linux Ubuntu 10.04. The experiments were carried out on different dense and sparse benchmark datasets.[4] Representative results are plot by Figure 3.

According to the obtained experimental results, interesting reduction rates are obtained whether *minsupp* varies or *minbond* varies. The $\mathcal{RCPR}$ representation is indeed proved to be a perfect cover of the $\mathcal{RCP}$ set. In fact, the size of $\mathcal{RCPR}$ is always smaller than that of $\mathcal{RCP}$ set over the entire range of the support and *bond* thresholds. For example, considering the dense MUSHROOM dataset for *minsupp* = *35*% and *minbond* = *0.15*: $|\mathcal{RCPR}|$ = *1, 810*, while $|\mathcal{RCP}|$ = *100, 156*. In this situation, $\mathcal{RCPR}$ offers a reduction reaching approximately *98*%. These results are obtained thanks to the non-injectivity of the closure operator f_{bond} which gathers into disjoint subsets, *i.e.*, f_{bond} *equivalence classes*, patterns that have the same characteristics. This process avoids mining redundant patterns. Note that, in this case, $|\mathcal{MRCP}|$ = *1, 412* and $|\mathcal{CRCP}|$ = *652*. Since the $\mathcal{RCPR}$ representation corresponds to the union without redundancy of the $\mathcal{MRCP}$ and $\mathcal{CRCP}$, we always have $|\mathcal{RCPR}| \leq |\mathcal{MRCP}| + |\mathcal{CRCP}|$.

We present in the following the application of the proposed representation to the context of intrusion detection.

[4] Available at *http://fimi.cs.helsinki.fi/data*.

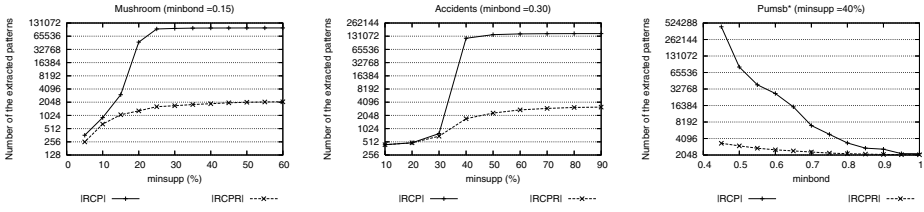


Fig. 3. Evaluation of the $\mathcal{RCPR}$ representation size w.r.t. the *minsupp* and *minbond* variations

6 Application to Intrusion Detection

We present in this section, the application of the $\mathcal{RCPR}$ representation in the design of an association rules based classifier. In fact, we used the $\mathcal{MRCP}$ and the $\mathcal{CRCP}$ sets, composing the $\mathcal{RCPR}$ representation, within the generation of the generic rare correlated rules of the form $Min \Rightarrow Closed \setminus Min$, with $Min \in \mathcal{MRCP}$ and $Closed \in \mathcal{CRCP}$.[5] Then, from the generated set of the generic rules, only the classification rules will be retained, *i.e.*, those having the label of the attack class in its conclusion part. After that, a dedicated classifier we designed is fed with these rules and has to perform the classification process and returns the detection rate for each attack class.

We present hereafter the application of our approach on the KDD 99 dataset.

6.1 Description of the KDD 99 Dataset

Each object of the KDD 99 dataset[6] represents a connection in the network data flow and is then labelled either normal or attack. KDD 99 defines 38 attacks categories partitioned into four `Attack` classes, which are DOS, PROBE, R2L and U2R, and one NORMAL class. The KDD 99 dataset contains 4, 940, 190 objects in the learning set and 41 input attributes for each connection. We propose in this work to consider 10% of the training set in the construction step of the classifier, containing 494, 019 objects. The learning set contains 79.20% (respectively, 0.83%, 19.65%, 0.22% and 0.10%) of DOS (respectively, PROBE, NORMAL, R2L and U2R).

6.2 Summary of Experimentations and Discussion of Obtained Results

Table 2 summarizes the obtained results, where AR and DR, respectively, denote "Association Rule" and "Detection Rate",[7] while *minconf* is the minimum threshold of the confidence measure [1]. In addition, by "Construction step", we mean that the step associated to the extraction of the $\mathcal{RCPR}$ representation while "Classification step"

[5] By "generic", it is meant that these rules are with minimal premises and maximal conclusions, w.r.t. set-inclusion.

[6] The KDD 99 dataset is available at the following link: *http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html*.

[7] Detection Rate = $\frac{NbrCcCx}{TotalNbrCx}$, with $NbrCcCx$ stands for the number of the correctly classified connections and $TotalNbrCx$ is equal to the whole number of the classified connections.

represents the step in which the classification association rules are derived starting from $\mathcal{RCPR}$ and applied for detecting intrusions.

We note that the highest value of the detection rate is achieved for the classes NORMAL and DOS. In fact, this is related to the high number of connections of these two classes. This confirms that our proposed approach presents interesting performances even when applied to voluminous datasets. We also remark that the detection rate varies from an attack class to another one. In fact, for the U2R class, this rate is relatively low when compared to the others classes.

To sum up, according to Table 2, the computational cost varies from one attack class to another one. It is also worth noting that, for all the classes, the construction step is much more time-consuming than the classification step. This can be explained by the fact that the extraction of the $\mathcal{RCPR}$ concise representation is an NP-hard problem since the localization of the associated two borders is a complex task.

Table 2. Evaluation of the rare correlated association rules for the KDD 99 dataset

Attack class	*minsupp* (%)	*minbond*	*minconf*	# of generic exact ARs	# of generic approximate ARs	# of generic ARs of classification	CPU Time (in seconds) Construction step	CPU Time (in seconds) Classification step
DOS	80	0.95	0.90	4	31	17	120	1
PROBE	60	0.70	0.90	232	561	15	55	1
NORMAL	85	0.95	0.95	0	10	3	393	15
R2L	80	0.90	0.70	2	368	1	1, 729	1
U2R	60	0.75	0.75	106	3	5	32	1

Furthermore, the results shown by Table 3 prove that the proposed rare correlated association rules are more competitive than the decision trees as well as the Bayesian networks [2]. In fact, our approach presents better results for the attack classes DOS, R2L and U2R than these two approaches. For the NORMAL class, the obtained results using our approach are close to those obtained with the decision trees. The Bayesian networks based approach presents better detection rate only for the PROBE attack class. The proposed rare correlated association rules then constitute an efficient classification tool when applied to the intrusion detection in a computer network.

Table 3. Comparison between the proposed rare correlated association rules based classifier versus the state of the art approaches

Attack class	Rare correlated generic ARs	Decision trees [2]	Bayesian networks [2]
DOS	**98.68**	97.24	96.65
PROBE	70.69	77.92	**88.33**
NORMAL	**100.00**	99.50	97.68
R2L	**81.52**	0.52	8.66
U2R	**38.46**	13.60	11.84

7 Conclusion and Future Works

We proposed in this paper a characterization of the $\mathcal{RCP}$ set of rare correlated patterns and we defined the new exact concise $\mathcal{RCPR}$ representation associated with this set. We then designed the RCPRMINER algorithm allowing an efficient extraction of this representation. The carried out experimental studies highlight interesting compactness rates

offered by $\mathcal{RCPR}$. The effectiveness of the proposed classification method, based on generic rare correlated association rules, has also been proved in the context of intrusion detection. Other avenues of future works concern the extraction of generalized association rules starting from rare correlated patterns and their use in real-life applications. In addition, we plan to extend our approach to other correlation measures [6,10,12,14] through classifying them into classes of measures sharing the same properties.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB 1994), Santiago, Chile, pp. 487–499 (1994)
2. Ben Amor, N., Benferhat, S., Elouedi, Z.: Naive bayes vs decision trees in intrusion detection systems. In: Proceedings of the ACM Symposium on Applied Computing (SAC 2004), Nicosia, Cyprus, pp. 420–424 (2004)
3. Boley, M., Gärtner, T.: On the Complexity of Constraint-Based Theory Extraction. In: Gama, J., Costa, V.S., Jorge, A.M., Brazdil, P.B. (eds.) DS 2009. LNCS, vol. 5808, pp. 92–106. Springer, Heidelberg (2009)
4. Boulicaut, J.F., Jeudy, B.: Constraint-based data mining. In: Data Mining and Knowledge Discovery Handbook, 2nd edn., pp. 339–354. Springer (2010)
5. Ganter, B., Wille, R.: Formal Concept Analysis. Springer (1999)
6. Kim, S., Barsky, M., Han, J.: Efficient Mining of Top Correlated Patterns Based on Null-Invariant Measures. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011, Part II. LNCS, vol. 6912, pp. 177–192. Springer, Heidelberg (2011)
7. Koh, Y.S., Rountree, N.: Rare Association Rule Mining and Knowledge Discovery: Technologies for Infrequent and Critical Event Detection. IGI Global Publisher (2010)
8. Kryszkiewicz, M.: Inferring Knowledge from Frequent Patterns. In: Bustard, D.W., Liu, W., Sterritt, R. (eds.) Soft-Ware 2002. LNCS, vol. 2311, pp. 247–262. Springer, Heidelberg (2002)
9. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery 3(1), 241–258 (1997)
10. Omiecinski, E.: Alternative interest measures for mining associations in databases. IEEE Transactions on Knowledge and Data Engineering 15(1), 57–69 (2003)
11. Pei, J., Han, J.: Constrained frequent pattern mining: a pattern-growth view. ACM-SIGKDD Explorations 4(1), 31–39 (2004)
12. Segond, M., Borgelt, C.: Item Set Mining Based on Cover Similarity. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part II. LNCS, vol. 6635, pp. 493–505. Springer, Heidelberg (2011)
13. Soulet, A., Crémilleux, B.: Adequate condensed representations of patterns. Data Mining and Knowledge Discovery 17(1), 94–110 (2008)
14. Surana, A., Kiran, R.U., Reddy, P.K.: Selecting a right interestingness measure for rare association rules. In: Proceedings of the 16th International Conference on Management of Data (COMAD 2010), Nagpur, India, pp. 115–124 (2010)
15. Taniar, D., Rahayu, W., Lee, V., Daly, O.: Exception rules in association rule mining. Applied Mathematics and Computation 205(2), 735–750 (2008)
16. Xiong, H., Tan, P.N., Kumar, V.: Hyperclique pattern discovery. Data Mining and Knowledge Discovery 13(2), 219–242 (2006)

Life Activity Modeling of News Event on Twitter Using Energy Function

Rong Lu, Zhiheng Xu, Yang Zhang, and Qing Yang

Institute of Automation, Chinese Academy of Science, Beijing, China
{rlu,zhxu,yzhang,qyang}@nlpr.ia.ac.cn

Abstract. This research is the first exploration on modeling life activity of news event on Twitter. We consider a news event as a natural life form, and use an energy function to evaluate its activity. A news event on Twitter becomes more active with a burst of tweets discussing it, and it fades away with time. These changes of the activity are well captured by the energy function. Then, we incorporate this energy function into the traditional single-pass clustering algorithm, and propose a more adaptive on-line news event detection method. A corpus of tweets which discuss news events was analyzed using our method. Experimental results show that our method not only compares favorably to those of other methods in official TDT measures like precision, recall etc., but also has better time and memory performance, which makes it more suitable for a real system.

Keywords: life activity modeling, energy function, Twitter, news event detection, single-pass clustering.

1 Introduction

Twitter is a very popular micro-blogging and social-networking service. More than 160 million users around the world are using it to remain socially connected to their friends, family members and co-workers[3]. It allows users to use a short text within a limit of 140 characters as their posts (also called ***tweets***) through many ways, including the mobile phone, the Web and text messaging tools[1] and so on. Twitter also employs a social-networking model called "following"[4], in which the user is allowed to follow any other users she wants to, without any permission or reciprocating by following her back. The one she follows is her ***friend***, and she is the ***follower***. Being a follower on Twitter means she receives all the updates from her friends[2].

More than a micro-blogging and social-networking service, Twitter is also like a news media. Many news outlets have accounts on Twitter, such as ABC, CNN, and New York Times. They use their accounts to report news, while many other users follow these accounts to subscribe news coverage. Up to now, New York Times has already have about 3 million followers. This news reporting and reading application is so popular on Twitter, because the short text makes the

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 73–84, 2012.

news easier to read, and the social-networking functionality makes it faster to diffuse and also provides a good interaction between users.

Unfortunately, Twitter grows too fast. The number of tweets per day is over 200 million. How to obtain the desired news information among the huge mass of tweets becomes a problem. Therefore, a real-time on-line news event detection system of Twitter is necessary. Usually, traditional single-pass algorithm[21] is used to handle this problem. However, there is an ambiguous place of the traditional single-pass algorithm. This algorithm clusters tweets into different clusters as different news events. But, it does not point out when to drop a news event out of the system memory, as there is no news tweet any more. And, this may cause it time consuming and memory exhausting for a real system.

In this paper, we implement the traditional single-pass clustering algorithm by modeling the life activity of news event. First, we use an energy function to model the life activity of a news event. We consider a news event on Twitter as a natural life. For a natural life, it eats different food containing different energy. It absorbs the energy by a certain transform ratio. Then, it grows old with time. Similarly, the tweet is food to a news event on Twitter. So the energy of a single tweet, an energy ***transferred factor*** and an energy ***decayed factor*** are introduced and integrated together as an energy function. The value of the energy function indicates the activity of a news event.

Then, we incorporate the energy function into the traditional single-pass algorithm. The threshold of the traditional single-pass algorithm is a constant. But, we use a variable to replace it. This variable threshold changes with the activity. We also add a time window to determine when a news event should be dropped out of the system memory. This time window changes with the activity of the news event, too.

The rest of this paper is organized as follows: In section 2, we give a review of related works. In Section 3, we describe the concepts and details of the energy function. Then, we incorporate it into the traditional single-pass clustering algorithm in Section 4. Section 5 reports the experiments and Section 6 concludes this paper.

2 Related Work

What is Twitter? Kwak, et al.[2] point out that Twitter is not only a social network, but something more akin to traditional news media. In its follower-following topology analysis, [2] has found a non-power-law follower distribution, a short effective diameter, and low reciprocity, which all mark a deviation from known characteristics of human social networks[10]. Actually, over 85% of trending topics on Twitter are headline or persistent news in nature. Java, et al. also give six main user intentions on Twitter in [11], and reporting news is one of them.

Now, reporting and reading news is one of the most important application on Twitter. As enormous amount of tweets are generated by the users everyday, it is necessary to detect and track news event automatically. Detecting and

Tracking new event was discussed in the project called Topic Detection and Tracking(TDT), which is a DARPA-sponsored activity to detect and track news events from streams of broadcast news stories.In general, clustering techniques are the major methods of TDT. Salton in [9] introduced hierarchical agglomerative clustering(HAC) method, and Yang, et al.[6] speed up the HAC by using the technique of bucketing and re-clustering. However, HAC is not very suitable for the time-ordered data collection. The other clustering method is single-pass clustering[21,5], which processes the input documents iteratively and chronologically. Ron and James discuss the implementation and evaluation of a on-line new event detection and tracking system using the single-pass clustering algorithm in [20]. They proposed a threshold model, which regarded exploiting temporal information would lead to improve the performance.

Besides, Chen, et al. proposed an aging theory to model life cycle of news events in order to improve the traditional single-pass clustering algorithm in [7]. This work is close to ours. However, we go much further. First, we clearly defined an energy function to evaluate the activity of a news event and give a iterative algorithm to solve the parameters. Second, we use the activity of a news event to determine the threshold of the single-pass clustering algorithm, and how long a news event should stay in the system memory. Third, [7] divides the news events into short-term and long-term events, while we treat all news events the same way. Finally, our work focuses on the stream of tweets instead of traditional news reports or stories.

With the rise of Twitter, researches of event detection on Twitter have already attracted some attention. Sakaki, et al. gave a real-time event detection algorithm, which monitors tweets to detect a target event like earthquake in [15]. In their work, they also found out the tweets of a news event follows an exponential distribution with time. [22] proposed a topic detection technique that permits to retrieve in real-time the most emergent topics. [23] introduced a method to collect, group, rank and track breaking news on Twitter, and developed an application called "Hotstream".

Another issue worthwhile to note is that the tweets are much shorter and noisier. The tweets stream is mixed by News events, Conversations[12], work communication[13], business information[14] and so on. [8] made an attempt to select the tweets that discussed news events only.

3 Modeling Life Activity Using Energy Function

In this section, the details of the energy function are described. We consider a news event on Twitter as a natural life form. To track its life activity, we use the concept of energy function. Like the endogenous fitness of an artificial life agent[16], the value of the energy function indicates the activity of a news event.

3.1 Definition of Energy Function

A news event on Twitter becomes popular with a burst of tweets discussing it and it fades away with time. A tweet discussed a news event is called **news**

tweet. A news tweet to a news event is like food to a natural life. It provides energy to the news event by a certain transform ratio. However, the life energy of the news event also diminishes with time as the same as natural life grows old. Figure 1 shows this life process.

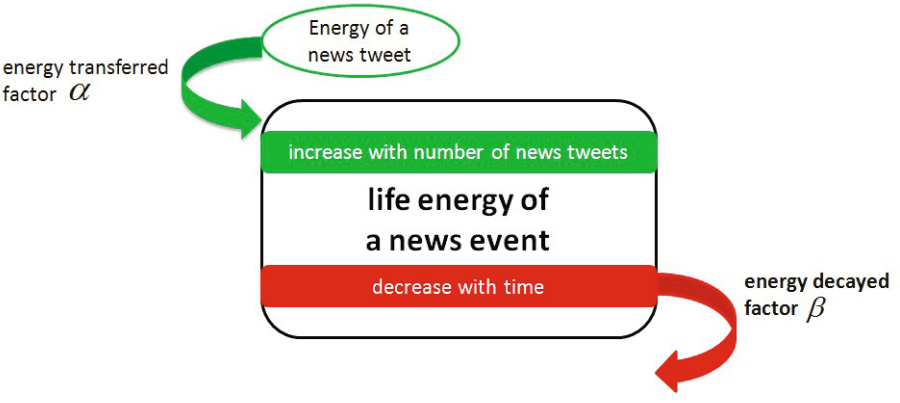


Fig. 1. Life process of a news event on Twitter

α is **energy transferred factor**, and β is **energy decayed factor**. Energy of a single news tweet is denoted by $itemEng$. Then we divide the whole life span(the time span from the first news tweet to the last one) of a news event into several successive and equal sized time slice. In time slice t, E_t represents the net energy that a news event obtains, including the energy absorb from news tweets and the energy lost with time in this time slice. A news tweet is denoted as d, and the news tweet set in this time slice is denoted as D_t. So, the net energy in this time slice t is defined as follows:

$$E_t = \sum_{d \in D_t} (\alpha \cdot itemEng(d)) - \beta \tag{1}$$

The total energy of a news event at the nth time slice is the sum of the net energy of all time slices before, so the **Energy Function** is:

$$E(n) = \sum_{t=1}^{n} E_t = \sum_{t=1}^{n} (\sum_{d \in D_t} (\alpha \cdot itemEng(d)) - \beta) \tag{2}$$

The value of $E(n)$ just indicates the activity of a news event at the nth time slice. It is easy to see $E(0) = 0$. If the news event has N time slices in total, another constraint of the energy function is:

$$E(N+1) = 0 \tag{3}$$

The meaning of Equation 3 is also obviously. When the news event is over, its energy value should turn to 0 again.

From Equation 2, when a burst tweets discuss a news event, the energy value increases, and the news event becomes more active. As time goes by, less and less tweets discuss the news event, the decayed factor β plays a more effective role, the energy value decreases, and the news event becomes less and less active.

3.2 Energy of A Single Tweet

As mentioned above, a news tweet to a news event is like food to natural life. Different food contains different nutrition. Similarly, different news tweets also contain different energy. The news tweet posted by more influential user will get more attention. If more users can read the news tweet, the news event will have more chance to become popular. It means a news tweet from a more influential user contains more energy.

As a social-networking service, the relationships between users construct a directed map, all users are the nodes of this graph. Researches[17,18,19] have already studied on how to measure the influence of a node in the directed graph or a person in a social network. One simple method of measuring the influence of users on Twitter is the **In-degree** method in[4], it measures the influence of a user by the number of her followers. This measurement currently also employed by Twitter and many other third-party services, such as *twitterholic.com* and *wefollow.com*.

Therefore, we also choose this measurement. f denotes the number of followers of a Twitter user, f_{max} denotes the maximum number of followers that a user has in our dataset. As a result, the influence of a Twitter user is defined as:

$$if_{user}(f) = \frac{\log(f)}{\log(f_{max})} \tag{4}$$

It is obvious that $0 \leq if_{user} \leq 1$.

The energy value of a single news tweet is denoted by (**itemEng**), which has already shown up in Equation 1 and 2. It is defined as:

$$itemEng(d) = \lambda_1 + \lambda_2 \cdot if_{user} \tag{5}$$

where, $0 \leq \lambda_1 \leq 1$, $0 \leq \lambda_2 \leq 1$, and $\lambda_1 + \lambda_2 = 1$.

3.3 Constant Growth and Decay

One particular case of the life cycle of a news event is constant growth and decay, which means, no matter how active the news event is, the energy transform ratio and the loss of energy are the same. In another word, the transferred factor α and the decay factor β are both constants. As a result, the energy function of Equation 2 can be reduced to a simpler form as:

$$E(n) = \sum_{t=1}^{n} E_t = \alpha \sum_{t=1}^{n} \sum_{d \in D_t} itemEng(d) - n\beta \tag{6}$$

There are two parameters α and β in Equation 6. We need two equations to solve them. Therefore, we let t_1, t_2 be two different time slices in the life span of a news event, and s_1, s_2 be the life energy at respective time slice. Then, we have:

$$\begin{cases} E(t_1) = \sum_{t=1}^{t_1} E_t = \alpha \sum_{t=1}^{t_1} \sum_{d \in D_t} itemEng(d) - t_1\beta = s_1 \\ E(t_2) = \sum_{t=1}^{t_2} E_t = \alpha \sum_{t=1}^{t_2} \sum_{d \in D_t} itemEng(d) - t_2\beta = s_2 \end{cases} \tag{7}$$

let

$$y(n) = \sum_{t=1}^{n} \sum_{d \in D_t} itemEng(d) \tag{8}$$

where $y(n)$ means the total energy the news tweets provide to the news event till nth time slice.

Then, solve Equation 7, we get α and β as follows:

$$\alpha = \frac{s_1 \cdot t_2 - s_2 \cdot t_1}{t_2 \cdot y(t_1) - t_1 \cdot y(t_2)} \tag{9}$$

and

$$\beta = \frac{s_1 \cdot y(t_2) - s_2 \cdot y(t_1)}{t_2 \cdot y(t_1) - t_1 \cdot y(t_2)} \tag{10}$$

4 Single-Pass Clustering with Energy Function

If news events were to be sought from a time-ordered static collection, one solution would be to use document clustering techniques [9,6] to cluster the collection, and then to return the document from each cluster containing the earliest timestamp. However, we are interested in the strict on-line data, which has real-time constraints and imposes a single-pass restriction over the incoming data stream of tweets. The traditional single-pass clustering algorithm for news event detection on Twitter, is described as follows:

1. Build a term vector representation for the tweets and news events. The term vector of a news event is represented by the geometric center of all term vectors of its news tweets.
2. Compare a new tweet against the previous news events in memory.
3. If the tweet does not trigger any previous news events by exceeding a threshold, flag the tweet as containing a new event, and add the news event into the memory.
4. If the tweet triggers an existing news event, flag the tweet to this news event, and update the term vector of the news event by recomputing the geometric center.

There are two shortcomings of the traditional single-pass clustering algorithm. First, the threshold of single-pass clustering method is a constant, which is not very reasonable. When a news event is hot and its energy value is high, there are a lot discussions on Twitter. Therefore, the threshold should turn smaller. So that, tweets about the same news event with different contents can be clustered into one news event. When the news event is dying, the news tweet is few. The threshold should turn bigger, in case of other news tweets are clustered in this news event.

Second, the traditional single-pass clustering algorithm does not mention how long a news event should stay in the memory. It wastes the system memory and also increases the time cost. Because, a new tweet still need to be compared to the dead news event, and it even has a small chance to be flagged to the dead news event. In a word, this shortcoming could reduce the performance of a real system in all aspects.

We modify the traditional single-pass clustering algorithm with energy function to conquer the two problems described above. For the first one, we make the threshold denoted by θ a variable, which changes with the energy value as follows:

$$\theta = \begin{cases} \theta_{max}, & E > E_2 \\ \dfrac{\theta_{max} - \theta_{min}}{E_2 - E_1} \times E + \dfrac{\theta_{min} \times E_2 - \theta_{max} \times E_1}{E_2 - E_1}, & E_1 \leq E \leq E_2 \\ \theta_{min}, & E < E_1 \end{cases} \tag{11}$$

E represents the energy value of a news event. θ changes linearly with E. And it has an upper bound θ_{max} and a lower bound θ_{min}, when E reaches E_2 and E_1.

For the second one, we check the time of the last tweet of every news event in the memory periodically. At the check point, the time is T, and the time of the last tweet of a news event e_i is T_i. A time window W is given. If $T - T_i > W$, e_i should be dropped out of the memory. This time window is also change with the energy value computed by the energy function as follows:

$$W = \begin{cases} w_{max}, & E > E_2 \\ \dfrac{w_{max} - w_{min}}{E_2 - E_1} \times E + \dfrac{w_{min} \times E_2 - w_{max} \times E_1}{E_2 - E_1}, & E_1 \leq E \leq E_2 \\ w_{min}, & E < E_1 \end{cases} \tag{12}$$

w_{max} and w_{min} are the upper bound and lower bound of W, respectively.

5 Experiments and Evaluation

Before experiments and evaluation, we give a brief description of the dataset for this research work. Then, we solve the energy transferred factor α and decayed factor β of the energy function of our dataset. Finally, our news event detection method is compared with others.

5.1 Data Preparation

For the purpose of this research work, we crawled 900 headlines of news reports from November 2, 2010 to January 10, 2011 through the RSS(Really Simple Syndication) of the Associated Press website[1]. For each news report, we crawled the tweets which matches all the words in the headline. Because all the news reports are published by one news outlet, they seldom talk about the same news event. We suppose that each news report represents an independent news event. Besides, there were other tweets we did not crawl also discussed the news event, only because they did not match all the words in the headlins of news report. However, the tweets we crawled can be regarded as a sample of the whole news tweets set. In this dataset, there are more than 400 thousand tweets in total, and these tweets were posted by more than 130 thousands users.

Then we divided the dataset into two sets. One is training set, which is used to train the energy transferred factor α, decayed Factor β, and threshold for the single-pass clustering algorithm. We randomly chose 259 news events as the training set. The rest 641 news events constitute the testing set, which is used for evaluation and comparison.

5.2 Training Energy Transferred Factor and Decayed Factor

In this subsection, an iterative algorithm is proposed to solve the energy transferred factor α and decayed factor β. Before that, one more point to add is the maximum energy value of each news event, which will be used to solve the energy transferred and decayed factors. We suppose the maximum energy of every news event is proportional to the its activity. As a result, for news events e_1 and e_2, they have c_1 and c_2 news tweets, their whole life span are l_1 and l_2 hours, and their maximum energy values are $max(e_1)$ and $max(e_2)$, the assumption can be expressed as:

$$\frac{max(e_1)}{c_1/{l_1}^{\mu}} = \frac{max(e_2)}{c_2/{l_2}^{\mu}} \tag{13}$$

where, $0 < \mu < 1$. If $\mu = 1$, $c_1/{l_1}^{\mu} = c_1/l_1$ is the average activity of news event e_1; If $\mu = 0$, $c_1/{l_1}^{\mu} = c_1$ is the total activity. So, when $0 < \mu < 1$, it can be regarded as the mixture of the average and the total activity. In our experiment, μ is set to 0.6.

Therefore, in the training set, if we set the maximum energy value of the news event e_{max} to 1.0, for other news event e, the maximum energy value is:

$$max(e) = \frac{c_e}{c_{e_{max}}} \times (\frac{l_{e_{max}}}{l_e})^{\mu} \tag{14}$$

The iterative algorithm to solve the energy transferred factor α and energy decayed factor β are described as follows:

[1] http://hosted.ap.org/lineups/TOPHEADS-rss_2.0.xml?SITE=ILMOL&SECTION=HOME

1. compute the maximum energy value $max(e)$ of every news event e in the training set by Equation 14.
2. compute the energy of every tweet in the training set by Equation 5, where λ_1 and λ_2 are set to 0.3 and 0.7, empirically.
3. for every news event e in the training set:
   (a) initialize $t_1 = 0.3l_e$, $s_1 = 0.7max(e)$, $t_2 = N + 1$, $s_2 = 0.0$, $t_{max} = 0$
   (b) repeat (c)(d)(e), until t_{max} does not change any more
   (c) compute α and β by Equation 8, 9, 10
   (d) find the maximum energy value and the time slice t_{max} using α and β above by Equation 2
   (e) reset $t_1 = t_{max}$, $s_1 = max(e)$, $t_2 = N + 1$, $s_2 = 0.0$
4. compute the average value of all α and β of all news event in the training set as the final results.

The final results are: $\alpha = 0.00110091$, $\beta = 0.00654238$. We also give all results of α and β for all news events in the training set in Figure 2.

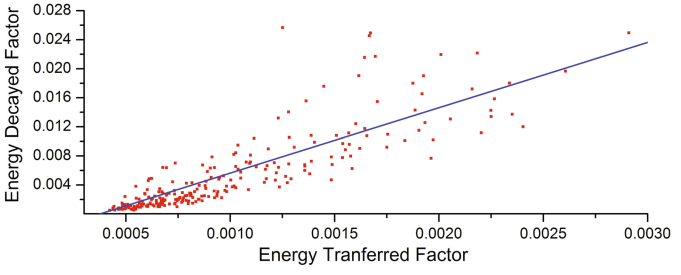


Fig. 2. Energy decayed factor vs. energy transferred factor

It is clear to see that there is a obvious linear correlation between the energy transferred factor and decayed factor. The main reason may be that all news events follow almost the same tweets distribution with time. [15] considered this distribution is an exponential distribution. So, using the average value of all energy transferred factors and decayed factors as the final results is appropriate.

5.3 News Event Detection Comparisons

In this experiment, our method(A) is compared to two other methods. The baseline method(B) is the traditional single-pass algorithm. The other is a fixed time-window single-pass clustering method(W). This fixed time-window method is a traditional single-pass clustering method added with a **fixed time window** W_{fixed}. In this method, it also check the news event in the memory periodically.

If there is no new web document for a news event more than W_{fixed} time, this simple modified method will consider the news event is over, and delete it from memory.

All the three methods group the tweets in the test set into several clusters. Five official TDT measures[5] including: precision(p), recall(r), miss(m), false alarm(f) and F1-measure($F1$) are used to evaluate the results of these three methods.

Table 1 shows the results. W4, W8, W12 are the fixed-time-window method with a fixed time window of 4, 8, 12 hours. In our method, the W_{min} and W_{max} in Equation 12 are set to 4 and 12 hours. So it is reasonable to compare our method with W4, W8 and W12.

Table 1. Results of TDT measures

	p	r	m	f	$F1$
B	0.877939	0.904714	0.095286	0.000285	0.891125
W4	0.947464	0.499588	0.500412	0.000065	0.654215
W8	0.941955	0.692312	0.307688	0.000096	0.798066
W12	0.932875	0.779466	0.220534	0.000124	0.849298
A	0.914556	0.876216	0.123784	0.000301	**0.894976**

In Table 1, all fixed time-window methods out-performance the baseline method a little in precision. However, their recalls are too low to accept. Thus, the baseline method has a better F1-measure than all fixed time-window methods. For our method, it achieves both reasonable precision and recall, which results in the best F1-measure of all methods.

Besides, our method also has acceptable time and memory performance. Figure 3 shows the time and memory performance of all three methods. The red lines are our method.

In Figure 3(a) and 3(b), The time cost of the traditional single-pass clustering method increases as the square of the number of tweets processed, while the fixed time-window method and our method are increase almost linearly. Our method is a little slower than the fixed time-window method. Because it needs a few more computational works of the changing threshold and time window, which is worthwhile. As there is a big improvement in official TDT measures, especially in recall.

In Figure 3(c) and 3(d), the memory cost is represented by the number of clusters in memory. The number of clusters of Traditional single-pass method increases linearly with the number of tweets processed, while the fixed time-window method and our method both fluctuate around a small constant.

Generally speaking, our method has the best results in the official TDT measures and it also has quite acceptable time and memory performance, which makes it suitable for a real system.

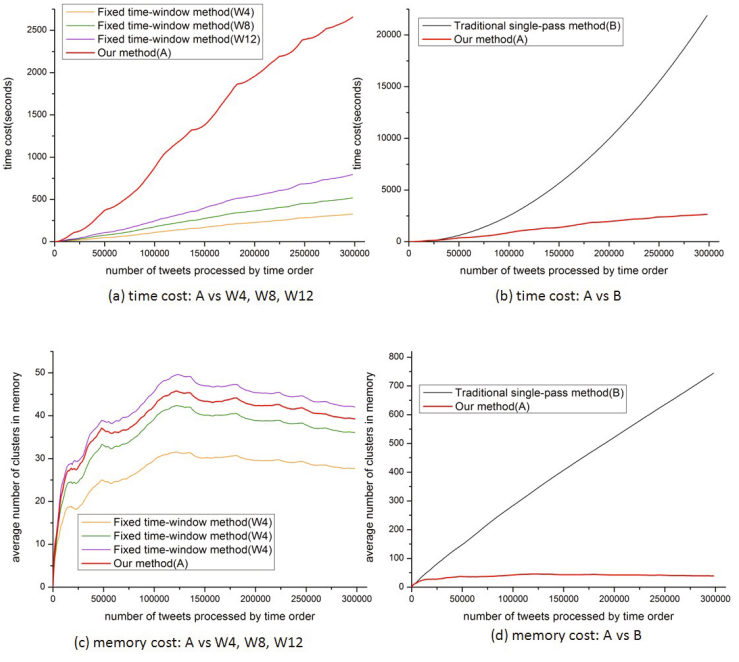


Fig. 3. Comparison of time and memory cost

6 Conclusions

In this paper, we report a novel news event detection method of Twitter. Experimental results show that it performances well not only in the official TDT measures, but also in time and memory cost.

Although the proposed method is quite good for a real system, there are still two major points needed to be improved. First, the energy transferred factor and decayed factor could also change with the energy value itself. When the news event is active, the energy transferred factor could be a little bigger, while the energy decayed factor could be a little smaller. Second, the user influence, which measures the energy of a single tweet, could use a more reliable and effective model. Moreover, our method can also be more generalized in other time sequential data mining, in the future.

References

1. Milstein, S., Chowdhury, A., Hochmuth, G., Lorica, B., Magoulas, R.: Twitter and the micro-messaging revolution: Communication, connections, and immediacy–140 characters at a time. O'Reilly Radar Report (2008)
2. Kwak, H., Lee, C., Park, H., Moon, S.: What is Twitter, a social network or a news media? In: WWW 2010, pp. 591–600 (2010)

3. Huberman, B.A., Romero, D.M., Wu, F.: Social networks that matter: Twitter under the microscope. Social Computing Laboratory, HP Labs (2008)
4. Weng, J., Lim, E.P., Jiang, J., He, Q.: TwitterRank: finding topic-sensitive influential twitterers. In: WSDM 2010, pp. 261–270 (2010)
5. Allan, J., Carbonell, J., Doddington, G., Yamron, J., Yang, Y.: Topic detection and tracking pilot study: Final report. In: Proc. of the DARPA Broadcast News Transcription and Understanding Workshop (1998)
6. Yang, Y., Pierce, T., Carbonell, J.: A study of retrospective and on-line event detection. In: SIGIR 1998, pp. 28–36 (1998)
7. Chen, C.C., Chen, Y.-T., Sun, Y., Chen, M.C.: Life Cycle Modeling of News Events Using Aging Theory. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) ECML 2003. LNCS (LNAI), vol. 2837, pp. 47–59. Springer, Heidelberg (2003)
8. Lu, R., Yang, Q.: Extracting News Topics from Microblogs based on Hidden topics discovering and Text Clustering. In: CCIR 2010, pp. 291–298 (2010)
9. Salton, G.: Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison-Wesley Longman Publishing Co., Inc. (1989)
10. Newman, M.E.J., Park, J.: Why social networks are different from other types of networks. Phys. Rev. E 68(3), 36–122 (2003)
11. Java, A., Song, X., Finin, T., Tseng, B.: Why we twitter: understanding microblogging usage and communities. In: Proc. of 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis, pp. 56–65 (2007)
12. Honey, C., Herring, S.C.: Beyond Microblogging: Conversation and Collaboration via Twitter. In: Proc. of the 42nd Hawaii International Conference on System Sciences, pp. 1–10 (2009)
13. Zhao, D., Rosson, M.B.: How and why people Twitter: the role that micro-blogging plays in informal communication at work. In: Proc. of the ACM 2009 International Conference on Supporting Group Work, pp. 243–252 (2009)
14. Coon, M., Reeves, B.: Social Media Marketing: Successful Case Studies of Businesses Using Facebook and YouTube With An In-Depth Look into the Business Use of Twitter (2010)
15. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes Twitter users: real-time event detection by social sensors. In: WWW 2010, pp. 851–860 (2010)
16. Menczer, F., Belew, R.K., Willuhn, W.: Artificial Life Applied to Adaptive Information Agents. In: AAAI 1995 (1995)
17. Kempe, D., Kleinberg, J., Tardos, É.: Influential Nodes in a Diffusion Model for Social Networks. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1127–1138. Springer, Heidelberg (2005)
18. Leavitt, A., Burchard, E., Fisher, D., Gillbert, S.: New approaches for analyzing influence on twitter. A publication of the Web Ecology project (2009)
19. Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Network and ISDN Systems, 107–117 (1998)
20. Papka, R., Allan, J.: On-Line New Event Detection using Single Pass Clustering. University of Massachusetts, Amherst (1998)
21. Van Rijsbergen, C.J.: Information Retereval, 2nd edn. Butterworths, Massachusetts (1979)
22. Cataldi, M., Di Caro, L., Schifanella, C.: Emerging topic detection on Twitter based on temporal and social terms evaluation. In: Proc. of the Tenth International Workshop on Multimedia Data Mining, pp. 1–10 (2010)
23. Phuvipadawat, S., Murata, T.: Breaking News Detection and Tracking in Twitter. In: International Conference on Web Intelligence and Intelligent Agent Technology, pp. 120–123 (2010)

Quantifying Reciprocity in Large Weighted Communication Networks

Leman Akoglu[1,3], Pedro O.S. Vaz de Melo[2,3], and Christos Faloutsos[1,3]

[1] Carnegie Mellon University, School of Computer Science
[2] Universidade Federal de Minas Gerais
[3] iLab, Heinz College
{lakoglu,christos}@cs.cmu.edu, olmo@dcc.ufmg.br

Abstract. If a friend called you 50 times last month, how many times did you call him back? Does the answer change if we ask about SMS, or e-mails? We want to quantify reciprocity between individuals in weighted networks, and we want to discover whether it depends on their topological features (like degree, or number of common neighbors). Here we answer these questions, by studying the call- and SMS records of *millions* of mobile phone users from a large city, with more than 0.5 *billion* phone calls and 60 *million* SMSs, exchanged over a period of six months. Our main contributions are: (1) We propose a novel distribution, the *Triple Power Law* (3PL), that fits the reciprocity behavior of all 3 datasets we study, with a better fit than older competitors, (2) 3PL is parsimonious; it has only three parameters and thus avoids over-fitting, (3) 3PL can spot anomalies, and we report the most surprising ones, in our real networks, (4) We observe that the *degree* of reciprocity between users is correlated with their local topological features; reciprocity is higher among mutual users with larger local network overlap and greater degree similarity.

1 Introduction

One of the important aspects in human relations is the reciprocity, a.k.a. mutuality. Reciprocity can be defined as the tendency towards forming mutual connections with one another by returning similar acts, such as email and phone calls. In a highly reciprocal relationship both parties share equal interest in keeping up their relationship, while in a relationship with low reciprocity, one person is much more active than the other.

It is important to understand the factors that play role in the formation of reciprocity as there exists evidence that reciprocal relationships are highly probable to persist in the future [6]. Also, [18] shows that reciprocity related behaviors provide good features for ranking and classification based methods for trust prediction. Reciprocity plays other important roles in social and communication networks. For example, if the network supports a propagation process, such as spreading of viruses in email networks or spreading of information and ideas in social networks, then the presence of mutual links clearly speeds up the propagation. Non-existence of reciprocal links can also reveal unwanted calls and emails in spam detection.

Despite its importance, reciprocity has remained an under-explored dynamic in networks. Most work in network science and social network analysis focus on node level degree distributions [5,10,14], communities [20,22,19], and triadic relations, such as

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 85–96, 2012.

clustering coefficients and triangle closures [12]. The study of *dyadic* relations [26] and the related *bivariate* distributions they introduce is, however, mostly overlooked, and thus is the focus of this paper. Our motivation is grouped into two topics:

M1. Modeling bivariate distributions in real data: Two vital components of understanding data at hand are studying the simple distributions in it and visualizing it [27]. The study of reciprocity introduces bivariate distributions, such as the distribution $\Pr(w_{ij}, w_{ji})$ of edge weights on mutual edges, where association between *two* quantitative variables needs to be explored. A vast majority of existing work focus on *univariate* distributions in real data such as power-laws [8], log-normals [4], and most recently DPLNs [21], however the study of *multivariate* distributions has limited focus.

In addition, visualization of multivariate data in 2D is hard and often misleading due to issues regarding over-plotting. More importantly, mere visualization does not provide a compact data representation as opposed to data modeling. Summarization via aggregate functions such as the average or the median loses a lot of information and is also not representative, especially for skewed distributions as found in real data.

Models, on the other hand, provide compact data representations by capturing the patterns in the data, and are ideal tools for applications like data compression and anomaly detection.

M2. A weighted approach to reciprocity: Traditional work [11] usually study reciprocity on directed, *unweighted* networks as a *global* feature which is quantified as the ratio of the number of mutual links pointing in both directions to the total number of links. Defining reciprocity in such an unweighted fashion, however, prevents understanding the *degree* of reciprocity between mutual dyads. In a weighted network, even though two nodes might have mutual links between them, the skewness and the magnitude of the weights associated with these links would contain more information about how much reciprocity is really there between these nodes. For example, in a phone call network the reciprocity between a mutual dyad where the parties make 80%-20% of their calls respectively is certainly different than that of a mutual dyad with 50%-50% share of their calls. In short, edge weights are crucial to study reciprocity as a property of each dyad rather than as a global feature of the entire network and give more insight into the *level* of mutuality.

In this paper, we analyze phone call and SMS records of 1.87 million mobile phone users from a large city collected over six months. The data consists of over half a billion phone calls and more than 60 million SMSs exchanged. Our contributions are:

1. We observe similar bivariate distributions $\Pr(w_{ij}, w_{ji})$ of mutual edge weights in the communication networks we study. We propose the Triple Power Law (3PL) function to model this observed pattern and show that 3PL fits the real data with millions of points very well. We statistically demonstrate that 3PL provides better fits than the well-known Bivariate Pareto and Bivariate Yule distributions. We also use 3PL to spot anomalies, such as a pair of users with low mutuality where one of the parties makes 99% of the calls during the entire working hours, non-stop.
2. We use weighted measures of reciprocity in order to quantify the degree of reciprocal relations and study the correlations between reciprocity and local topological features among user pairs. Our results suggest that mutual users with larger local network overlap and higher degree similarity exhibit greater reciprocity.

2 Related Work

Bivariate Distributions in Real Data: A vast majority of existing work focus on *univariate* distributions in real data such as power-laws, Pareto distributions and so on [17]. For example, the degree distribution has been found to obey a power-law in many real graphs such as the Internet Autonomous Systems graph [10], the WWW link graph [1], and others [5,7]. Additional power laws seem to govern the popularity of posts in citation networks, which drops over time, with power law exponent of -1 for paper citations and -1.5 for blog posts [14].

A recent comprehensive study [8] on power-law distributions in empirical data shows that while power-laws exist in many graphs, deviations from a pure power-law are also observed. Those deviations usually appear in the form of exponential cut-offs and log-normals. Similar deviations were also observed in [2] where the electric power-grid graph in a specific region in California as well as airport networks were found to exhibit power-law distributions with exponential cut-offs.

Other deviations from power-laws continue. Discrete Gaussian Exponential (DGX) [4] was shown to provide good fits to distributions in a variety of real world data sets such as the Internet click-stream data and usage data from a mobile phone operator. Most recently, [21] studied several phone call networks and proposed a new distribution called the Double Pareto Log-Normal (DPLN) that was used to separately model the per-user number of call partners, number of calls and number of minutes. Other related work on explaining and modeling the behaviour of phone network users include [6,9,16,23].

While univariate distributions are used to model the distribution of a specific quantity x, for example the number of calls of users, bivariate distributions are used to model the association and co-variation between two quantitative variables x_1 and x_2. Association is based on how two variables simultaneously change together, for example the total number of calls with respect to the number of call partners of users.

Unlike univariate distributions, the multivariate distributions have mostly been studied theoretically in mathematics and statistics [3]. On the other hand, analysis of especially skewed multivariate distributions *in real data* has attracted much less focus. Existing work includes [28], which uses the bivariate log-normal distribution to describe the joint distributions of flood peaks and volumes, as well as flood volumes and durations. Also, [15] studies the drought in the state of Nebraska and models the duration and severity, proportion and inter-arrival time, and duration and magnitude of drought with bivariate Pareto distributions.

Reciprocity in Unweighted Networks: Previous studies usually consider reciprocity as a global metric of a given directed network where reciprocity is quantified as $r=\frac{L^{\leftrightarrow}}{L}$, the ratio of the number of mutual links $L^{\leftrightarrow}$ pointing in both directions to the total number of links L. By definition, r=1 for a purely bidirectional network (e.g. collaboration networks) and r=0 for a purely unidirectional network (e.g. citation networks).

There are two issues with this definition. First, it depends on the density of the network; reciprocity is larger in a network with larger link density. Second, this definition treats the graph as unweighted, and thus fails to quantify the *degree* of reciprocity between mutual dyads. [11] combines this classical definition with the network density into a single measure which tackles the first problem, however the new measure still remains a global, unweighted metric and does not allow to study the degree of reciprocity.

3 Data Description

In this work, we study anonymous mobile communication records of millions of users collected over a period of six months, December 1, 2007 through May 31, 2008. The data set contains both phone call and SMS interactions.

From the whole six months' of activity, we build three networks, CALL-N, CALL-D and SMS, in which nodes represent users and directed edges represent phone call and SMS interactions between these users. CALL-N is a who-calls-whom network with edge weights denoting (1) total number of phone calls, CALL-D is the same who-calls-whom network with edge weights denoting (2) total duration of phone calls (aggregated in minutes), and SMS is a who-texts-whom network with edge weights denoting (3) total number of SMSs. Table 1 gives the data statistics. Global unweighted reciprocity is r=0.84 for CALL, and r=0.24 for SMS.

Table 1. Data statistics. The number of nodes N, the number of directed edges E, and the total weight W in the mutual and non-mutual CALL and SMS networks.

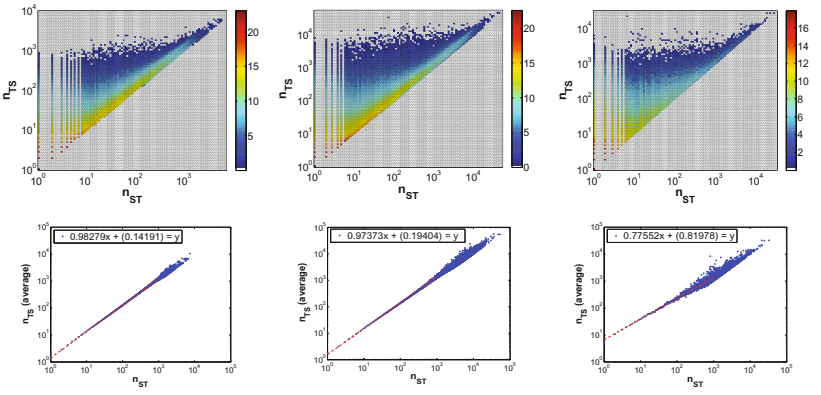
Network	N	E	W_N	$W_D(min)$	**Network**	N	E	W_{SMS}
CALL	1,87M	49,50M	483,7M	915x10^6	SMS	1,87M	8,80M	60,5M
CALL(mutual)	1,75M	41,84M	468,7M	885x10^6	SMS(mutual)	0,58M	2,10M	46,6M

4 Proposed Model: 3PL

Given a network of users with *mutual*, *weighted* edges between them, say CALL-N, and given two users i and j in the network, is there a relation between the number of calls i makes to j (w_{ij}) and the number of calls j makes to i (w_{ji})? In this section, we want to understand the association between the weights on the reciprocal edges in human communication networks and study their distribution $\Pr(w_{ij}, w_{ji})$ across mutual dyads. Since we study the pair-wise joint distribution, the order of the weights do not matter. Thus, to ease notation, we will denote the smaller of these weights as n_{ST} (for weight from Silent-to-Talkative) and the larger as n_{TS}, and will study $\Pr(n_{ST}, n_{TS})$.

Figure 1(top-row) shows the weights n_{TS} versus n_{ST} for all the reciprocal edges in (from left to right) CALL-N, CALL-D, and SMS. Each dot in the plots corresponds to a pair of mutual edges. Since there could be several pairs with the same (n_{ST}, n_{TS}) weights, the regular scatter plot of the reciprocal edge weights would result in over-plotting. Therefore, in order to make the densities of the regions clear, we show the *heatmap* of the scatter plots where colors represent the magnitude of volume (red means high volume and blue means low volume).

In Figure 1, we observe that most of the points are concentrated (1) around the origin and (2) along the diagonal for all three networks. Concentration around the origin, for example in CALL-N, suggests that the vast majority of people make only a few phone calls with $n_{ST}, n_{TS} < 10$, and much fewer people make many phone calls, which points to skewness. In addition, concentration along the diagonal indicates that mutual people call each other mostly in a balanced fashion with $n_{ST} \approx n_{TS}$. Notice that similar arguments hold for CALL-D and SMS.



(a) CALL-N (b) CALL-D (c) SMS

Fig. 1. (top-row) Scatter plot heatmaps: total weight n_{ST} (Silent to Talkative) vs the reverse, n_{TS}, in log scales. Visualization by scatter plots suffers from over-plotting. Heatmaps color-code dense regions but do not have compact representations or formulas. Figures are best viewed in color; red points represent denser regions. The counts are in $\log_2$ scale. (bottom-row) Aggregation by average: summarization and data aggregation, e.g. averaging, loses a lot of information.

Even though heatmaps reveal similar patterns in all the three networks, mere visualization does not provide compact representations for our data. One way to go around this issue is to do data summarization. For example, Figure 1 (bottom-row) shows how n_{TS} changes with n_{ST} *on average*. The least square fit of the data points in log-log scales then provides a mathematical representation of the data. Data summarization by means of an aggregate function such as the average, however, loses a lot of information about the actual distribution: in our example, the slope of the least square fit in CALL-N is close to 1, which suggests that n_{TS} is equal to n_{ST} on average, and does not provide any information for the deviations. This issue arises mostly because aggregation by the average is not a good representative, especially for skewed distributions.

Given our observation that the distribution of reciprocal edge weights (n_{ST}, n_{TS}) follows a similar pattern across all three networks, how can we model the observed distributions? Since neither visualization nor aggregation qualify for compact data representation, we propose to formulate the distributions with the following bivariate functional form $\Pr(n_{ST}, n_{TS})$, which we call the Triple Power-Law (3PL) function.

Proposed Model 1 (Triple Power-Law (3PL)). *In human communication networks, the distribution* $\Pr(n_{ST}, n_{TS})$ *of mutual edge weights* n_{ST} *and* n_{TS} *(*n_{ST} *being the smaller) follows a Triple Power-Law in the following form*

$$\Pr(n_{ST}, n_{TS}; \alpha, \beta, \gamma) \quad \propto \boxed{\frac{n_{ST}^{-\alpha} n_{TS}^{-\beta} (n_{TS} - n_{ST} + 1)^{-\gamma}}{Z(\alpha, \beta, \gamma)}}, \alpha > 0, \beta > 0, \gamma > 0, \textit{ and}$$

$$n_{TS} \geq n_{ST} > 0,\ Z(\alpha, \beta, \gamma) = \sum_{n_{ST}=1}^{M} \sum_{n_{TS}=n_{ST}}^{M} n_{ST}^{-\alpha} n_{TS}^{-\beta} (n_{TS} - n_{ST} + 1)^{-\gamma}.$$

where Z is the normalization constant and M is a very large integer.

Next we elaborate on the intuition behind the exponents α, β and γ.

Intuition behind the β Exponent: 3PL is the 2D extension of the "rich-get-richer" phenomenon; people who make many phone calls will continue making even more, and even longer ones, leading to skewed, power-law-like distributions. The β exponent is the skewness of the main component, the number n_{TS} of phone-calls from 'talkative' to 'silent'. High β means more skewed distribution; β=0 is roughly uniform distribution. As we show in Figure 1, there are many people who make only a few (and short) phone calls and only a few people who make many (and long) phone calls. Visually, the vast majority of people who make only a few phone calls are represented with the high density (dark red) regions around the origin in all three networks.

Intuition behind the α Exponent: Similarly, this indicates the skewness for n_{ST}, the number of silent-to-talkative phone-calls. High value of α means high skewness, while α close to zero means uniformity. Notice that $\alpha \approx 0$ for our real phone-call datasets (see Figure 2).

Intuition behind the γ Exponent: It captures the skewness in asymmetry. High γ means that large asymmetries are improbable. This is the case in all our real datasets. For example, in addition to the origin in Figure 1(a), the regions along the diagonal also have high densities. These regions correspond to mutual pairs with about equal interaction in both directions. This suggests that humans tend to reciprocate their communications. 3PL also captures this observation; notice that the probability is higher for n_{TS} close to n_{ST} and drops for larger inequality $(n_{TS} - n_{ST})$ as a power-law with exponent γ.

4.1 Comparison of 3PL to Competing Models

In this section, we compare our model with two well-known parametric distributions for skewed bivariate data, the Bivariate Pareto [13] and the Bivariate Yule [25]. Their functional forms are given as two alternative competitor models as follows.

Competitor Model 1 (Bivariate Pareto)

$$f_{X_1,X_2}(x_1,x_2) = k(k+1)(ab)^{k+1}(ax_1+bx_2+ab)^{-k-2}, x_1,x_2,a,b,k>0.$$

Competitor Model 2 (Bivariate Yule)

$$f_{X_1,X_2}(x_1,x_2) = \frac{\rho_{(2)}(x_1+x_2)!}{(\rho+1)_{(x_1+x_2+2)}}, x_1,x_2,\rho>0; \alpha_{(\beta)} = \Gamma(\alpha+\beta)/\Gamma(\alpha), \alpha>0, \beta \in R.$$

We use maximum likelihood estimation to fit the parameters of each model for each of our three networks. In Figure 2, we report the best-fit parameters as well as the corresponding data log likelihood scores (the higher, the better). Notice that for CALL-N and CALL-D the 3PL achieves higher data likelihood than both Bivariate Pareto and Bivariate Yule. On the other hand, for SMS, the data likelihood scores of all three models are about the same; with Bivariate Pareto giving a slightly higher score.

The simple sign of the difference between the log likelihoods (log likelihood ratio $\mathcal{R}$), however, does not on its own show conclusively that one distribution is better than the other as it is subject to statistical fluctuation. If its true value over many independent

data sets drawn from the same distribution is close to zero, then the fluctuations can easily change its sign and thus the results of the comparison cannot be trusted. In order to make a firm judgement in favor of 3PL, we need to show that the difference between the log likelihoods is sufficiently large and that it could not be the result of a chance fluctuation. To do so, we need to know the standard deviation σ on $\mathcal{R}$, which we estimate from our data using the method proposed in [24].

	CALL-N	CALL-D	SMS
Triple Power Law (3PL)			
α	1e-06	1e-06	0.8120
β	2.0703	1.8670	1.5896
γ	0.8204	0.9650	0.3005
Loglikelihood	-7.55e+07	-8.88e+07	-5.41e+06
Bivariate Pareto			
k	0.7407	0.7657	0.7862
a	0.2119	0.5723	0.7097
b	10e+05	1.25e+04	0.7553
Loglikelihood	-7.77e+07	-9.26e+07	-5.39e+06
z	803.73	975.75	-41.06
p	0	0	0
Bivariate Yule			
ρ	1.11e-16	5.55e-17	1e-06
Loglikelihood	-8.59e+07	-10.00e+07	-5.41e+06
z	2.14e+03	1.93e+03	1.49
p	0	0	0.03

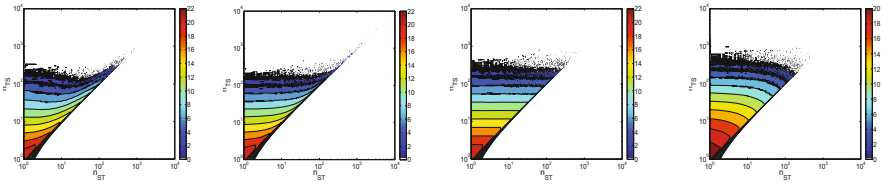
Fig. 2. Maximum likelihood parameters estimated for 3PL, Bivariate Pareto and the Bivariate Yule and data log-likelihoods obtained with the best-fit parameters. We also give the normalized log likelihood ratios z and the corresponding p-values. A positive (and large) z value indicates that 3PL is favored over the alternative. A small p-value confirms the significance of the result. Notice that 3PL provides significantly better fits to CALL and is as good as its competitors for SMS.

In Figure 2, we report the normalized log likelihood ratio denoted by $z = \mathcal{R}/\sqrt{2n}\sigma$, where n is the total number of data points (number of mutual edge pairs in our case). A positive z value indicates that the 3PL model is truly favored over the alternative. We also show the corresponding p-value, $p = erfc(z)$, where *erfc* is the complementary Gaussian error function. It gives an estimate of the probability that we measured a given value of $\mathcal{R}$ when the true value of $\mathcal{R}$ is close to zero (and thus cannot be trusted). Therefore, a small p value shows that the value of $\mathcal{R}$ is unlikely to be a chance result and its sign can be trusted.

Notice that the magnitude of z for CALL-N and CALL-D is quite large, which makes the p-value zero and shows that 3PL is a significantly better fit for those data sets. On the other hand, z is relatively much smaller for SMS, therefore we conclude that 3PL provides as good of a fit as its competitors for this data set. Note that the number of mutual edge pairs n in SMS ($\approx$1 million) is much smaller compared to that of the call networks ($\approx$21 million) (Table 1). It is worth emphasizing that difference, because the bivariate pattern of reciprocity might reveal itself better in larger data sets, and it would be interesting to see whether 3PL provides a better fit for SMS when more data samples become available.

Next, we demonstrate also visually that 3PL provides a better fit to the real data than its competitors. To this end, having estimated the model parameters for all three models, we generated synthetic data sets with the same number of samples as in each of our networks. We show the corresponding plots for CALL-N in Figure 3 (a) for real

data, and synthetic data generated by (b) 3PL, (c) Bivariate Pareto, and (d) Bivariate Yule. We notice that the simulated data distribution from 3PL looks more realistic than its two competitors. Similar results for CALL-D and SMS are omitted for brevity.



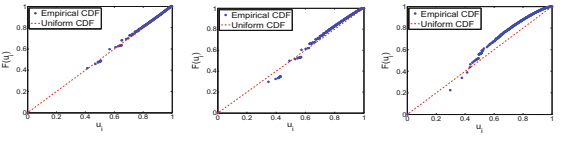
(a) CALL-N (real data) (b) 3PL (synthetic) (c) Biv. Pareto (synthetic) (d) Biv. Yule (synthetic)

Fig. 3. Contour-maps for the scatter plot n_{TS} versus n_{ST} in CALL-N (a) for real data, and synthetic data simulated from (b) 3PL, (c) Bivariate Pareto and (d) Bivariate Yule functions using the best-fit parameters. Notice that synthetic data generated by 3PL looks more similar to the real data than its competitors also visually. Counts are in $\log_2$ scale. Figures are best viewed in color.

4.2 Goodness of Fit

The likelihood ratio test is used to compare two models to determine which one provides a *better* fit to a given data. However, as we mentioned in the previous section, it cannot directly show when both competing models are poor fits to the data; it can only tell which is the least bad. Therefore, in addition to showing that 3PL provides a better (or as good) fit than its two competitors, we also need to demonstrate that it indeed provides a good fit itself.

A general class of tests for goodness of fit work by transforming the data points $(x_{1,i}, x_{2,i})$ according to a cumulative distribution function (CDF) F as $u_i = F(x_{1,i}, x_{2,i})$ for $\forall i, 1 \leq i \leq n$. One can show that if F is the correct CDF for the data, u_i should be *uniformly* distributed (derivation follows from basic probability theory). That is, if the CDF $\hat{F}$ estimated from our model is approximately correct, the empirical CDF of the $\hat{u}_i = \hat{F}(x_{1,i}, x_{2,i})$ should be approximately a straight line from $(0, 0)$ to $(1, 1)$.



(a) CALL-N (b) CALL-D (c) SMS

Fig. 4. Distribution of $\hat{u}_i = \hat{F}(x_{1,i}, x_{2,i})$ for all data points i according to cumulative distribution function (CDF) $\hat{F}$ estimated from our 3PL model. An approximately uniform distribution of $\hat{u}_i$ shows that 3PL provides a good fit to real data.

For each of our three data sets, we generate synthetic data drawn from our 3PL function with the corresponding estimated best-fit parameters. Then, we compute $\hat{u}_i = \hat{F}(x_{1,i}, x_{2,i})$ for all the data points in each of the data sets, where $\hat{F}$ is the estimated CDF from each synthetic data. In Figure 4, we show the CDF of $\hat{u}_i$ as well as the CDF for a perfect uniform distribution. Notice that the distribution of $\hat{u}_i$ is almost uniform for CALL-N and CALL-D, and quite close to the uniform for SMS. This corroborates our case that our model provides a good approximate to the correct CDF of our data sets, and thus indeed provides a *good* fit.

4.3 3PL at Work

There exist at least three levels at which we can make use of parametric statistical models for real data: (1) *as data summary*: compact mathematical representation, data reduction; (2) *as simulators*: generative tools for synthetic data; (3) *in anomaly detection*: probability density estimation.

In Figure 5(a), we show top 100 pairs in CALL-D with lowest 3PL likelihood (marked with triangles). Figure 5(b) shows the local neighborhood of one of the pairs, say A and B (marked with circles in (a)). We notice low mutuality; A initiated 99% of the calls in return to less than 2 hours total duration of calls B made. Further inspection revealed constant daily activity by A, including weekends, with about 7 hours call duration per day on average, starting at around 9am in the morning until around 5-8pm in the evening. It is also surprising that all these calls are addressed to the same contact, B. While for privacy reasons, we cannot fully tell the scenario behind this behavior, this proves to be an interesting case for the service operator to further look into. Other interesting anomalous observations are omitted for brevity.

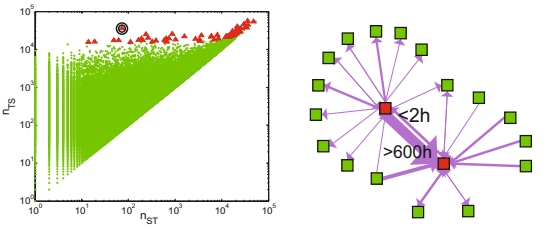


Fig. 5. (a) Least likely 100 points by 3PL (shown with triangles). (b) Local neighborhood of one mutual pair detected as an outlier (marked with circles). Edge thickness is proportional to edge weight.

5 Reciprocity and Local Network Topology

Given that person i calls person j w_{ij} times and person j calls person i w_{ji} times, what is the *degree* of reciprocity between them? In this section, we discuss several *weighted* metrics that quantify reciprocity between a given mutual pair. Later, we study the relationship between reciprocity among mutual pairs and their topological similarity.

5.1 Weighted Reciprocity Metrics

Three metrics we considered in this work to quantify the "similarity" or "balance" of weights w_{ij} and w_{ji} are (1) *Ratio* $r = \frac{\min(w_{ij},w_{ji})}{\max(w_{ij},w_{ji})} \in [0,1]$, (2) *Coherence* $c = \frac{2\sqrt{w_{ij}w_{ji}}}{(w_{ij}+w_{ji})} \in [0,1]$ (geometric mean divided by the arithmetic mean of the edge weights), and (3)*Entropy* $e = -p_{ij}\log_2(p_{ij}) - p_{ji}\log_2(p_{ji}) \in [0,1]$, where $p_{ij} = \frac{w_{ij}}{(w_{ij}+w_{ji})}$ and $p_{ji} = 1 - p_{ij}$. All these metrics are equal to 0 for the (non-mutual) pairs where one of the edge weights is 0, and equal to 1 when the edge weights are equal. Although these metrics are good at capturing the *balance* of the edge weights, they fail to capture the *volume* of the weights. For example, human would score $(w_{ji}{=}100, w_{ij}{=}100)$ higher than $(w_{ji}{=}1, w_{ij}{=}1)$, whereas all the metrics above would treat them as equal.

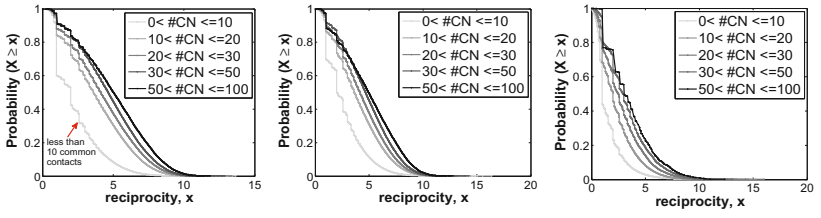
Therefore, we propose to multiply these metrics by the logarithm of the total weight, such that the reciprocity score consists of both a "balance" as well as a "volume" term. In

the rest of this section, we use the *weighted* ratio $r_w = \frac{\min(w_{ij},w_{ji})}{\max(w_{ij},w_{ji})} \log(w_{ij}+w_{ji})$ as the reciprocity measure in our experiments. The results are similar for the other weighted metrics, c_w and e_w.

5.2 Reciprocity and Network Overlap

Here, we want to understand whether there is a relation between the local network overlap (local density) and reciprocity between mutual pairs. Local network overlap of two nodes is simply the number of common neighbors they have in the network.

In Figure 6, we show the cumulative distribution of reciprocity separately for different ranges of overlap. The figures suggest that people with more common contacts tend to exhibit higher reciprocity, both in their SMS and phone call interactions.



(a) CALL-N (b) CALL-D (c) SMS

Fig. 6. Complementary cumulative distribution of reciprocity for different ranges of local network overlap (number of Common Neighbors). Notice that the more the number of common contacts, the higher the reciprocity.

5.3 Reciprocity and Degree Similarity

Next, we investigate the relation between the degree similarity (degree assortativity) and reciprocity. In Figure 7, we show the heatmap for the average reciprocity among pairs with respective degrees d_i and d_j for CALL-N (similar figures for other networks are omitted for brevity). The heatmap plot suggests that two people with more similar number of contacts exhibit larger reciprocity; notice the increase in reciprocity with increasing d_j for fixed d_i (from bottom to diagonal, towards degree similarity) and then the drop from diagonal to the right, towards degree dissimilarity.

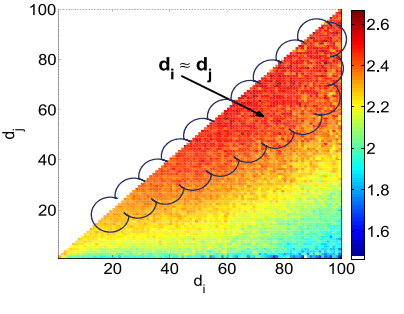


Fig. 7. Average reciprocity among dyads with degrees (d_i, d_j) in CALL-N

6 Conclusions

In this paper, we analyze more than 0.5 *billion* phone call and 60 *million* SMS records of millions of mobile phone users over six months; and study reciprocity; the distribution

and strength of mutual relations in weighted human communication networks. Our main contributions and findings are the following:

- **Patterns in joint pdf $\Pr(\mathbf{w_{ij}}, \mathbf{w_{ji}})$:** We find that the joint distribution $\Pr(w_{ij}, w_{ji})$ of the weights on mutual edges in mobile communication networks of users follow a bivariate pattern for all three types of weights; number of phone calls, duration of phone calls and number of SMSs. More specifically, the data points concentrate (1) around the origin as well as (2) along the diagonal in the scatter plot of w_{ij} versus w_{ji}. Observation (1) suggests a power-law like distribution in the amount of interactions; e.g., many people with few calls and only a few people with many calls. Observation (2) indicates that human communications are mostly reciprocal.
- **New model (3PL) for the joint pdf $\Pr(\mathbf{w_{ij}}, \mathbf{w_{ji}})$:** We propose the Triple Power Law (3PL) bivariate function to model this joint distribution. Our goodness of fit tests show that 3PL can model the observed distributions with more than 20 *million* mutual edge pairs quite well. We statistically demonstrate that it provides better fits than two other well-known bivariate distributions for skewed data, the Bivariate Pareto and the Bivariate Yule.
- **3PL at work:** 3PL provides a compact as well as a sparse data representation with only three parameters. We also show how to exploit 3PL to detect anomalies. Our case studies successfully reveal suspicious mutual interactions that agree with human intuition.
- **Weighted reciprocity:** Lastly, we take a weighted network approach and use weighted metrics to quantify the *degree* of reciprocity in human interactions. We observe that reciprocity is higher (1) for mutual pairs with larger local network overlap, that is, people with more common friends; and (2) for mutual pairs with larger degree-similarity, that is, people with similar number of contacts.

Acknowledgements. Research was sponsored by the National Science Foundation under Grant No. IIS1017415 and the Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053. It is continuing through participation in the Anomaly Detection at Multiple Scales (ADAMS) program sponsored by the U.S. Defense Advanced Research Projects Agency (DARPA) under Agreements No. W911NF-11-C-0200 and W911NF-11-C-0088. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, of the National Science Foundation, of the U.S. Government, or any other funding parties. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

1. Albert, R., Barabasi, A.-L.: Emergence of scaling in random networks. Science, 509–512 (1999)
2. Amaral, L.A.N., Scala, A., Barthélémy, M., Stanley, H.E.: Classes of small-world networks. Proceeding of the National Academy of Sciences (2000)
3. Arnold, B.C.: Bivariate distributions with pareto conditionals. Statistics & Probability Letters 5(4), 263–266 (1987)

4. Bi, Z., Faloutsos, C., Korn, F.: The "DGX" distribution for mining massive, skewed data. In: KDD (2001)
5. Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., Wiener, J.: Graph structure in the web: experiments and models. In: WWW (2000)
6. Cesar, C.R.-S., Hidalgo, A.: The dynamics of a mobile phone network. Physica A: Statistical Mechanics and its Applications 387(12), 3017–3024 (2008)
7. Chakrabarti, D., Zhan, Y., Faloutsos, C.: R-MAT: A recursive model for graph mining. In: SDM (2004)
8. Clauset, A., Shalizi, C.R., Newman, M.E.J.: Power-law distributions in empirical data. SIAM Rev. 51(4), 661–703 (2009)
9. Eagle, N., Pentland, A., Lazer, D.: Inferring social network structure using mobile phone data. Proceedings of the National Academy of Sciences (PNAS) 106, 15274–15278 (2009)
10. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. In: SIGCOMM, pp. 251–262 (August-September 1999)
11. Garlaschelli, D., Loffredo, M.I.: Patterns of Link Reciprocity in Directed Networks. Phys. Rev. Lett. 93, 268701 (2004)
12. Granovetter, M.: The strength of weak ties. Amer. Jour. of Sociology 78, 1360–1380 (1973)
13. Kotz, S., Balakrishnan, N., Johnson, N.L.: Continuous multivariate distributions, 2nd edn. Models and Applications, vol. 1 (2000)
14. Leskovec, J., McGlohon, M., Faloutsos, C., Glance, N., Hurst, M.: Cascading behavior in large blog graphs: Patterns and a model. In: SDM (2007)
15. Nadarajah, S.: A bivariate pareto model for drought. Stochastic Environmental Research and Risk Assessment 23, 811–822 (2009)
16. Nanavati, A.A., Gurumurthy, S., Das, G., Chakraborty, D., Dasgupta, K., Mukherjea, S., Joshi, A.: On the structural properties of massive telecom call graphs: findings and implications. In: CIKM 2006, pp. 435–444. ACM, New York (2006)
17. Newman, M.E.J.: Power laws, Pareto distributions and Zipf's law. Contemporary Physics 46(5), 323–351 (2005)
18. Nguyen, V.-A., Lim, E.-P., Tan, H.-H., Jiang, J., Sun, A.: Do you trust to get trust? a study of trust reciprocity behaviors and reciprocal trust prediction. In: SDM, pp. 72–83 (2010)
19. Nussbaum, R., Esfahanian, A.-H., Tan, P.-N.: Clustering social networks using distance-preserving subgraphs. In: ASONAM (2010)
20. Satuluri, V., Parthasarathy, S.: Scalable graph clustering using stochastic flows: applications to community discovery. In: KDD, pp. 737–746 (2009)
21. Seshadri, M., Machiraju, S., Sridharan, A., Bolot, J., Faloutsos, C., Leskovec, J.: Mobile call graphs: beyond power-law and lognormal distributions. In: KDD, pp. 596–604 (2008)
22. Tantipathananandh, C., Berger-Wolf, T., Kempe, D.: A framework for community identification in dynamic social networks. In: KDD, pp. 717–726 (2007)
23. Vaz de Melo, P.O.S., Akoglu, L., Faloutsos, C., Loureiro, A.A.F.: Surprising Patterns for the Call Duration Distribution of Mobile Phone Users. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010, Part III. LNCS, vol. 6323, pp. 354–369. Springer, Heidelberg (2010)
24. Vuong, Q.H.: Likelihood ratio tests for model selection and non-nested hypotheses. Econometrica 57, 307–333 (1989)
25. Xekalaki, E.: The bivariate yule distribution and some of its properties. Statistics 17(2), 311–317 (1986)
26. Xiang, R., Neville, J., Rogati, M.: Modeling relationship strength in online social networks. In: WWW, pp. 981–990 (2010)
27. Yang, X., Asur, S., Parthasarathy, S., Mehta, S.: A visual-analytic toolkit for dynamic interaction graphs. In: KDD, pp. 1016–1024 (2008)
28. Yue, S.: The bivariate lognormal distribution to model a multivariate flood episode. Hydrological Processes 14, 2575–2588 (2000)

Hierarchical Graph Summarization: Leveraging Hybrid Information through Visible and Invisible Linkage

Rui Yan[1], Zi Yuan[2], Xiaojun Wan[3], Yan Zhang[1,⋆], and Xiaoming Li[1]

[1] School of Electronics Engineering and Computer Science, Peking University, China
[2] School of Computer Science and Engineering, Beihang University, China
[3] Institute of Computer Science and Technology, Peking University, China
{r.yan,wanxiaojun,lxm}@pku.edu.cn,
ziyuan@cse.buaa.edu.cn, zhy@cis.pku.edu.cn

Abstract. Graph-based ranking algorithm has been recently exploited for summarization by using sentence-to-sentence relationships. Given a document set with linkage information to summarize, different sentences belong to different documents or clusters (either *visible* cluster via anchor texts or *invisible* cluster by semantics), which enables a hierarchical structure. It is challenging and interesting to investigate the impacts and weights of source documents/clusters: sentence from important ones are deemed more salient than the others. This paper aims to integrate three types of hierarchical linkage into traditional graph-based methods by proposing Hierarchical Graph Summarization (HGS). We utilize a hierarchical language model to measure the sentence relationships in HGS. We develop experimental systems to compare 5 rival algorithms on 4 instinctively different datasets which amount to 5197 documents. Performance comparisons between different system-generated summaries and manually created ones by human editors demonstrate the effectiveness of our approach in ROUGE metrics.

Keywords: Summarization, Hierarchical Graph, Visible and Invisible Linkage.

1 Introduction

In the era of information explosion, people need new information to update their knowledge whilst information on Web is updating extremely fast. Multi-document summarization has been proposed to address such dilemma by producing a summary delivering the majority of information content from a document corpus, and the short summary is necessarily helpful to facilitate users to quickly understand the large number of documents. Automated multi-document summarization has drawn much attention in recent years. In the communities of information retrieval and natural language processing, a series of conferences on automatic text summarization have advanced the summarization techniques and produced a couple of experimental online systems.

Graph-based ranking algorithms have been recently exploited for summarization by making use of sentence-to-sentence relationships and played an important role with the exponential document growth on the Web. In general, traditional graph summarization

⋆ Corresponding author.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 97–108, 2012.

utilizes plain linkage among sentences without considering higher-level information beyond the sentence-level information, which is insufficient. Given a document set with linkage information to summarize, different sentences belong to different documents and clusters, either clustered by **visible** linkage (e.g., anchor texts) or **invisible** linkage (e.g., semantic cohesion), which enables a hierarchical text structure. It is challenging and interesting to investigate the impacts and weights of source documents/clusters: different documents and clusters usually have different importance for users to understand the document set. Sentence from important documents/clusters are deemed more salient than the trivial ones. In brief, simultaneous consideration of three-layer hierarchical linkage has not been investigated under a unified framework.

In order to address above insufficiency, we aim to model these three levels of hierarchical linkage, i.e., sentence-to-sentence, sentence-to-document and document-to-cluster relationships, into traditional graph-based summarization, and we name this approach as Hierarchical Graph Summarization (HGS). We propose a hierarchical language model to measure the sentence relationships for the ranking process in HGS. Document/cluster-level information through visible and invisible linkage is used for smoothing: neighboring text information is proved to be useful [11]. We will first investigate the presence of visible and invisible linkage for clustering.

Visible Linkage. A web document is connected to other web documents by explicit links via anchor texts, which are denoted as visible linkage.

Invisible Linkage. A web document is connected to other web documents through implicit semantic coherence, denoted as invisible linkage.

The contributions of this paper are as follows:

• The **1st contribution** is to utilize the instinctively explicit linkage among web documents, which is a natural understanding of enormous web data organization. We distinguish such visible linkage from invisible linkage by semantic cohesion and utilize both information into clustering.

• The **2nd contribution** is to incorporate a three-level hierarchical linkage structure into a unified language smoothing model, which is used to measure sentence relationships by utilizing both document-level and cluster-level information simultaneously.

We start by reviewing previous work in Section 2. In Section 3 we describe the basic graph summarization and describe our proposed HGS in Section 4. We conduct empirical evaluations in Section 5, including performance comparisons and result discussion. Finally we draw conclusions in Section 6.

2 Related Work

Multi-document summarization (MDS) has drawn much attention in recent years. In general, MDS can either be extractive or abstractive. The former assigns salient scores to semantic units (e.g. sentences, paragraphs) of documents indicating the importance and then extracts top ranked ones, while the latter demands information fusion(e.g. sentence compression and reformulation). Here we focus on extractive summarization.

To date, various extraction-based methods have been proposed for generic multi-document summarization. MEAD [3] is an implementation of the centroid-based

method that scores sentences based on features such as cluster centroids, position, and TF.IDF, etc. NeATS [6] adds new features such as topic signature and term clustering to select important content. Themes (or topics, clusters) in documents have been discovered and used for sentence selection [10,14,13].

Most recently, the graph-based ranking methods have been proposed to rank sentences/passages based on "votes" or "recommendations" between each other. TextRank [9] and LexPageRank [2] use algorithms similar to PageRank and HITS to compute sentence importance. Cluster information such as document-level information has been incorporated in the graph model to better evaluate sentences [12].

Generally, summarization considers content characteristics such as coverage, diversity [1,17,5], and all these characteristics require a calculation of sentence linkage measurement. To the best of our knowledge, currently, neither the instinctively visible linkage of anchor texts from web document organizations is utilized for summarization, nor the three-layer hierarchical linkage has been investigated simultaneously in a unified language model to measure sentence relationships. HGS approach can naturally and simultaneously take into account these two advantages in graph-based summarization.

3 Basic Graph Summarization

The basic graph summarization is essentially a way of deciding the importance of a vertex within a linkage graph based on global information recursively drawn from the entire graph, using the Markov Random Walk Model (MRW). The basic idea is that of "voting" or "recommendation" between the vertices, where each vertex is a sentence. A link between two vertices is considered as a vote cast from one vertex to the other vertex. The score associated with a vertex is determined by the votes that are cast for it, and the score of the vertices casting these votes.

Formally, given a document set D, let $G = (V, E)$ be a graph to reflect the relationships between sentences in the document set, as shown in Figure 1 (Part A). V is the set of vertices and each vertex s_i in V is a sentence in the document set. E is the set of edges, which is a subset of $V \times V$. Each edge e_{ij} in E is associated with an affinity weight $f(s_i \rightarrow s_j)$ between sentences s_i and s_j ($i \neq j$). Sentence s is generated from the language model Θ_s. The affinity weight is measured by Kullback-Leibler divergence of s_i and s_j, contained in a decreasing logistic function $\mathcal{L}(x) = \frac{1}{1+e^x}$ to map the distance into interval [0,1] as proposed in [16,17]. That is, $f(s_i \rightarrow s_j) = \mathcal{L}(D_{KL}(s_j||s_i))$, where $D_{KL}(s_j||s_i)$ is:

$$D_{KL}(s_j||s_i) = \sum_{w \in W} p(w|\Theta_{s_j}) \log \frac{p(w|\Theta_{s_j})}{p(w|\Theta_{s_i})} \tag{1}$$

W is the set of words in our vocabulary and w denotes a word. The language model of Θ_s will be discussed in details later. If Θ_{s_i} and Θ_{s_j} are very close, the KL-divergence would be small and $f(s_i \rightarrow s_j)$ would be high, which intuitively makes sense.

Given $f(s_i \rightarrow s_j)$, the transition probability from s_i to s_j is then defined by normalizing the corresponding affinity weight as follows.

$$p(s_i \rightarrow s_j) = \begin{cases} \frac{f(s_i \rightarrow s_j)}{\sum_{k=1}^{|V|} f(s_i \rightarrow s_k)}, & if \sum f \neq 0 \\ 0, & otherwise \end{cases} \tag{2}$$

Note that $p(s_i \rightarrow s_j)$ is asymmetric and it measures the affinity from s_i to s_j. We let $f(s_i \rightarrow s_i) = 0$ to avoid self transition. We use the row-normalized matrix $M = [M_{ij}]_{|V|\times|V|}$ where $M_{ij} = p(s_i \rightarrow s_j)$ to describe G with each entry corresponding to the transition probability and all zero elements are replaced by a smoothing factor empirically set to $1/|V|$.

Based on the matrix *M*, the saliency score $\Psi(s_i)$ for sentence s_i can be deduced from those of all other sentences linked with it and it can be formulated in a recursive form as in the PageRank algorithm as follows:

$$\Psi(s_i) = \mu \cdot \sum_{all\ j \neq i} \Psi(s_j) \cdot M_{ji} + \frac{1-\mu}{|V|} \tag{3}$$

For implementation, the initial scores of all sentences are set to 1 and the iteration algorithm in Equation (3) is adopted to compute the new scores of the sentences. Usually the convergence of the iteration algorithm is achieved when the difference between the scores computed at two successive iterations for any sentences falls below a given threshold (0.0001 in this study). μ is the damping factor usually set to 0.85, as in the PageRank algorithm. We then apply the Maximum Marginal Relevance (MMR) mechanism for redundancy removal, similar to the method used in [11].

We see that according to the KL-divergence scoring method, our main tasks are to estimate Θ_s. Since s can be regarded as a short document, we can use any standard method to estimate Θ_s. Here, we use Dirichlet prior smoothing [18] to estimate Θ_s as follows:

$$p(w|\Theta_s) = \frac{c(w,s) + \mu_s \cdot p(w|B)}{|s| + \mu_s} = \frac{c(w,s)}{|s| + \mu_s} + \frac{\mu_s}{|s| + \mu_s} \cdot p(w|B) \tag{4}$$

where $|s|$ is the length of s, $c(w,s)$ is the count of word w in s, $p(w|B)$ is a background model used as smoothing factor. Generally $p(w|B)$ is estimated by the whole document set D, i.e., using $\frac{c(w,D)}{\sum_{w' \in W} c(w',D)}$. μ_s is the smoothing parameter.

However, note that as the length of a sentence is very short, smoothing is critical for addressing the term sparseness problem for sentences. The globalized smoothing from the whole corpus is coarse-grained. Therefore, we move on to estimate the fine-grained $p(w|\Theta_s)$ from multiple-layers by the hierarchical graph summarization.

4 Hierarchical Graph Summarization

4.1 Overview

In the basic graph summarization, all sentences are indistinguishable, i.e., the sentences are treated uniformly. As we mentioned in Section 1, there may be many factors that can have impact on the importance analysis of the sentences. This study aims to examine the impact of hierarchical linkage on graph summarization, by incorporating sentence-to-document relationship, as well as visible and invisible document clustering.

Besides **1)** the basic pair-wise *sentence-to-sentence* relationship, the hierarchical graph includes **2)** *sentence-to-document* relationship, **3)** *document-to-cluster* relationship from *visible* linkage and **4)** *document-to-cluster* relationship from *invisible* linkage by semantic clustering. **We number these four types of linkage correspondingly**

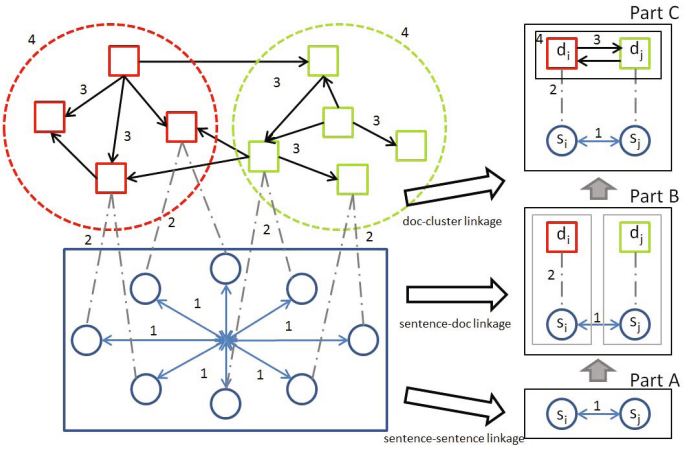


Fig. 1. Illustration of the hierarchical linkage graph. A circle denotes a sentence and a square denotes a document. Different lines denote different types of linkage, which are marked with Number **1-4**. Some lines are omitted due to the space limits.

in Fig. 1. As can be seen, the lowest layer is just the traditional link graph between sentences that has been well studied in previous work. The upper layer represents the documents. The dashed lines between these two layers indicate the conditional influence between the sentences and the documents: a link is established when the sentence is from the document. Documents are connected due to visible lines by anchor text arrows and are also grouped by invisible semantic clusters.

4.2 Incorporating Hierarchical Linkage

Sentence-to-Document Links. To incorporate the document-level information and the sentence-to-document relationship, the document-based graph model is proposed based on the two-layer link graph including both sentences and documents in Fig. 1.(Part B): the language model of sentence s is smoothed by the source document.

Visible Document-to-Cluster Links. Web documents are linked to each other through anchor texts and we keep such structural information. We start "walking" from a particular web document to all connected web documents until all linked documents are visited. These documents are clustered together as visible clusters, and the document within the visible cluster forms a visible document-to-cluster relationship.

Invisible Document-to-Cluster Links. Web documents can be clustered according to their semantic coherence, and the distance is calculated by the standard cosine similarity measurement. We use the popular clustering algorithms of K-means to produce the invisible semantic cluster. Given a document set, it is hard to predict the actual cluster number, and thus we empirically set the number k of expected clusters as k=$\sqrt{|D|}$, where $|D|$ is the number of documents.

Linkage Integration. After we introduce three types of hierarchical links, the estimation of the background language model Θ_B should be based on the source document

and source cluster where the sentence comes from, according to [8], the background model can be now written as:

$$p(w|B) = \frac{c(w,d) + \mu_c p(w|C)}{|d| + \mu_c} = \frac{c(w,d)}{|d| + \mu_c} + \frac{\mu_c}{|d| + \mu_c} \cdot p(w|C) \tag{5}$$

We take Equation (5) into Equation (4) and obtain the final representation:

$$\begin{aligned} p(w|\Theta_s) =& \frac{c(w,s)}{|s| + \mu_s} \cdot \frac{|s|}{|s|} + \frac{\mu_s}{|s| + \mu_s} \cdot \left(\frac{c(w,d)}{|d| + \mu_c} \cdot \frac{|d|}{|d|} + \frac{\mu_c}{|d| + \mu_c} \cdot p(w|C) \right) \\ =& \frac{|s|}{|s| + \mu_s} \cdot p(w|s) + \frac{\mu_s |d|}{(|s| + \mu_s)(|d| + \mu_c)} \cdot p(w|d) \\ &+ \frac{\mu_s \mu_c}{(|s| + \mu_s)(|d| + \mu_c)} \cdot p(w|C) \end{aligned} \tag{6}$$

μ_c can be interpreted as our confidence on the prior of how cluster information weighs. Thus setting μ_c=$|d|$ means that we put equal weights on the document-level and the cluster-level information. μ_c=0 yields no consideration of cluster-level information and μ_s=0 yields simple consideration of plain sentence relationships.

After simple calculation, we notice that the sum of all coefficients in Equation (6) equals to 1, and hence we change Equation (6) into a more concise format of

$$p(w|\Theta_s) = \alpha \cdot p(w|s) + \beta \cdot p(w|d) + \gamma \cdot p(w|C) \tag{7}$$

α, β, γ all belong to [0,1] and $\alpha + \beta + \gamma$=1. The cluster representation of $p(w|C)$ can be rewritten as a combination of visible cluster $p(w|C_v)$ and invisible cluster $p(w|C_{iv})$ controlled by λ:

$$p(w|C) = \lambda \cdot p(w|C_{iv}) + (1 - \lambda) \cdot p(w|C_v) \tag{8}$$

Special Cases:

(1) β=0 and γ=0: only plain relationship between two sentences are considered;
(2) $\beta \neq 0$, γ=0: plain linkage and document-to-sentence relationship included;
(3) $\gamma \neq 0$, λ=0 means no invisible clustering impact from visible linkage;
(4) $\gamma \neq 0$, λ=1 means no visible clustering impact from invisible linkage.

4.3 Estimation of Document/Cluster Importance

Documents and clusters are not equally important. Our assumption is that the sentences in an important document or cluster should be ranked higher and more likely to be chosen into the summary. The importance of documents (or clusters) is measured the relevance to the whole corpus. We examine such impact by incorporating the document importance and cluster importance into calculation of sentence linkage and ranking.

The function $\pi(d)$ aims to evaluate the importance of document d in the document set D. The following two methods are developed to evaluate the document importance.

π_{kl}: It uses the transformed KL-Divergence value between the document d and the whole document set D as the importance score of the document:

$$\pi_{kl}(d) = \mathcal{L}(D_{KL}(d||D)) \tag{9}$$

π_{pr}: It constructs a weighted graph between documents and uses the PageRank algorithm to compute the rank scores of the documents as the importance scores of the documents. The link structure among documents is established by the inherent visible linkage. The equation for iterative computation is the same with Equation (3).

The function $\phi(C)$ evaluates the importance of cluster C (both visible and invisible) in the document set D. Similarly we have two methods to evaluate cluster weights.

ϕ_{kl}: It uses the transformed KL-Divergence value between the cluster C and the whole document set D as the importance score of the cluster:

$$\phi_{kl}(C) = \mathcal{L}(D_{KL}(C||D)) \tag{10}$$

ϕ_{pr}: We add the PageRank scores of all the documents within the cluster C, i.e.,

$$\phi_{pr}(C) = \sum_{d \in C} \pi_{pr}(d) \tag{11}$$

By incorporating document and cluster importance, Equation (7) can be rewritten as Equation (12), substituting the unweighted $p(w|d)$ and $p(w|C)$. $p(w|\Theta_s)$ is estimated for all sentences and applied into Equation (1), (2), (3) to calculate the hierarchical sentence relationships and to rank sentences within the multiple-layer graph.

$$p(w|\Theta_s) = \alpha \cdot p(w|s) + \beta \cdot [\pi(d)p(w|d)] + \gamma \cdot [\phi(C)p(w|C)] \tag{12}$$

5 Experiments and Evaluation

5.1 Dataset

We use the data in [16] to test HGS on the real world datasets, which amounts to 5197 documents from various major news sites (such as BBC, CNN and Xinhua News, etc.). Our data includes 4 subjects, and each belongs to a different category of Rule of Interpretation (ROI) [4]. Reference summaries are created by editors [16].

Table 1. Detailed basic information of 4 datasets

Subjects	#Sentences	#Documents	#Visible Links	#RefSum (Avg. Length)
1.Influenza A	115026	2557	5108	5 (83)
2.BP Oil Spill	63021	1468	2493	6 (76)
3.Haiti Earthquake	12073	247	115	2 (32)
4.Michael Jackson Death	37819	925	1627	3 (64)

5.2 Evaluation Metrics

The ROUGE measure is widely used for evaluation [7]: the DUC contests usually officially employ ROUGE for automatic summarization evaluation. In ROUGE evaluation, the summarization quality is measured by counting the number of overlapping units, such as N-gram, word sequences, and word pairs between the candidate summaries CS

and the reference summaries RS. There are several kinds of ROUGE metrics, of which the most important one is ROUGE-N with 3 sub-metrics: precision, recall and F-score.

$$\text{ROUGE-N-R} = \frac{\sum_{\text{S}\in\text{RS}} \sum_{\text{N-gram}\in\text{S}} \text{Count}_{\text{match}}(\text{N-gram})}{\sum_{\text{S}\in\text{RS}} \sum_{\text{N-gram}\in\text{S}} \text{Count (N-gram)}}$$

$$\text{ROUGE-N-P} = \frac{\sum_{\text{S}\in\text{CS}} \sum_{\text{N-gram}\in\text{S}} \text{Count}_{\text{match}}(\text{N-gram})}{\sum_{\text{S}\in\text{CS}} \sum_{\text{N-gram}\in\text{S}} \text{Count (N-gram)}}$$

$$\text{ROUGE-N-F} = \frac{2 \times \text{ROUGE-N-P} \times \text{ROUGE-N-R}}{\text{ROUGE-N-P} + \text{ROUGE-N-R}}$$

S denotes a summary. N in these metrics stands for the length of N-gram and N-gram$\in$RS denotes the N-grams in reference summary while N-gram$\in$CS denotes the N-grams in the candidate summary. $\text{Count}_{\text{match}}$(N-gram) is the maximum number of N-gram in the candidate summary and in the set of reference summaries. $\text{Count}_{\text{(N-gram)}}$ is the number of N-grams in reference summaries or candidate summaries.

According to [7], among all sub-metrics, unigram-based ROUGE (ROUGE-1) has been shown to agree with human judgment most and bigram-based ROUGE (ROUGE-2) fits summarization well. We report three ROUGE F-measure scores: ROUGE-1, ROUGE-2, and ROUGE-W, where ROUGE-W is based on the weighted longest common subsequence. The weight W is set to be 1.2 in our experiments by ROUGE package (version 1.55). The higher the ROUGE scores, the similar the two summaries are.

5.3 Algorithms for Comparison

Pre-processing. Given a collection of documents, we first decompose them into sentences. Then the stop-words are removed and words stemming is performed. After these steps, we implement the following widely used summarization algorithms as baseline systems. They are designed for traditional summarization without hierarchical linkage. For fairness we conduct the same preprocessing for all algorithms.

Random: The method selects sentences randomly for each document collection.

Centroid: The method applies MEAD algorithm [3] to extract sentences according to the following parameters: centroid value, positional value, and first-sentence overlap.

GMDS: The plain graph MDS proposed by [11] first constructs a sentence connectivity graph based on cosine similarity and then selects important sentences based on the concept of eigenvector centrality.

PGMDS: Wan et al. present a two-layer pair-wise graph summarization methods in [12], utilizing sentence-to-sentence and sentence-to-document linkage without a consideration of simultaneous document-to-cluster links.

HGS: HGS is an algorithm with three-layer hierarchical linkage information and at the same time, both visible and invisible document clustering are performed.

RefSum: As we have used separate reference summaries from human evaluators, we not only provide ROUGE evaluations of the competing systems but also of the reference summaries against each other, which provides a good indicator of not only the upper bound ROUGE score that any system could achieve.

5.4 Overall Performance Comparison

We use a **cross validation** manner among 4 datasets, i.e., to train parameters on one subject set and to examine the performance on the others. After 4 training-testing processes, we take the average F-score performance in terms of ROUGE-1, ROUGE-2 and ROUGE-W on all sets. The details are listed in Tables 2∼5.

Table 2. Overall performance comparison on *Influenza A*. ROI* category: Science.

Systems	R-1	R-2	R-W	95%-conf.
RefSum	0.491	0.112	0.159	0.44958
Random	0.197	0.039	0.081	0.75694
Centroid	0.241	0.050	0.094	0.45073
GMDS	0.252	0.059	0.098	0.33269
PGMDS	**0.303**	0.060	0.099	0.53123
HGS	0.298	**0.063**	**0.101**	0.53459

Table 3. Overall performance comparison on *BP Oil Leak*. ROI category: Accidents.

Systems	R-1	R-2	R-W	95%-conf.
RefSum	0.517	0.135	0.183	0.48618
Random	0.202	0.041	0.096	0.64406
Centroid	0.259	0.052	0.098	0.34743
GMDS	0.267	0.057	0.102	0.43877
PGMDS	0.273	**0.061**	0.107	0.77245
HGS	**0.299**	0.058	**0.111**	0.39236

Table 4. Overall performance comparison on *Haiti Earthquake*. ROI category: Disasters.

Systems	R-1	R-2	R-W	95%-conf.
RefSum	0.528	0.139	0.167	0.30450
Random	0.206	0.043	0.093	0.75694
Centroid	0.252	0.050	0.099	0.43045
GMDS	0.251	0.058	0.098	0.33694
PGMDS	0.275	0.055	0.106	0.64198
HGS	**0.307**	**0.060**	**0.115**	0.67312

Table 5. Overall performance comparison on *Jackson Death*. ROI category: Legal Cases.

Systems	R-1	R-2	R-W	95%-conf.
RefSum	0.482	0.115	0.163	0.47052
Random	0.189	0.039	0.084	0.52426
Centroid	0.255	0.048	0.089	0.21045
GMDS	0.267	0.055	0.095	0.30070
PGMDS	0.281	**0.063**	0.107	0.67825
HGS	**0.294**	0.059	**0.113**	0.42148

*ROI: news categorization defined by Linguistic Data Consortium (http://www.ldc.upenn.edu/projects/tdt4/annotation).

From the results in Table 2 to Table 5, we have following observations:

• Generally Random has the worst performance.

• The results of Centroid are better than those of Random, mainly because the Centroid method takes into account positional value and first-sentence overlap, which facilitate main aspects summarization. However, the flat clustering-based summarization is proved to be less useful [15].

• The GMDS system outperforms centroid-based summarization methods. This is due to the fact that PageRank-based framework ranks the sentence using eigenvector centrality which implicitly accounts for information subsumption among all sentences.

• In general, the PGMDS algorithm outperforms GMDS system. It indicates that the two-layer hierarchical summarization is more useful than plain graph summarization and richer linkage structure indeed facilitates graph summarization.

• HGS under our proposed framework outperforms baselines, indicating that the overall properties we use for three layers of hierarchical linkage are beneficial for summarization tasks.

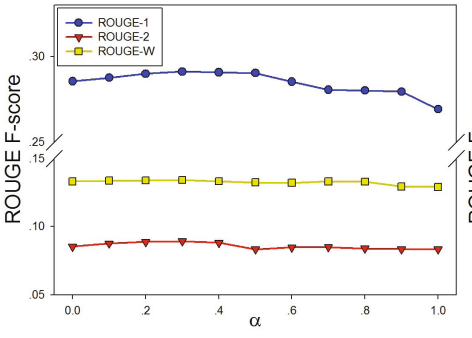


Fig. 2. α: weight of sentence-to-sentence links

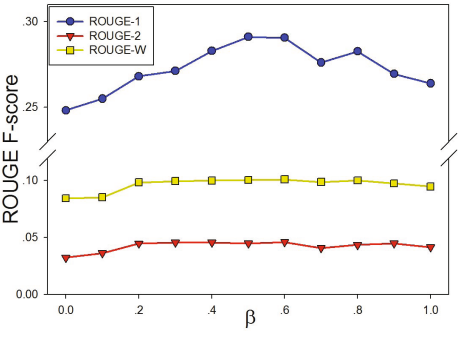


Fig. 3. β: weight of sentence-to-document links

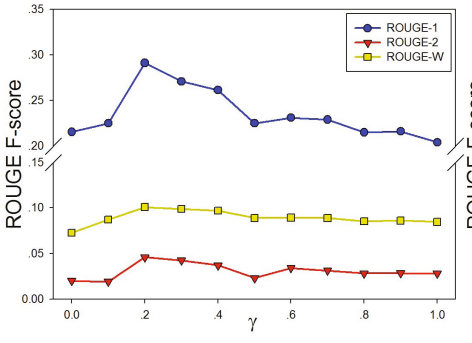


Fig. 4. γ: weight of document-to-cluster links

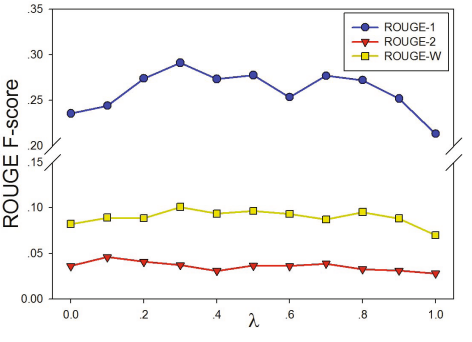


Fig. 5. λ: tradeoff of visible/invisible cluster

Having proved the effectiveness of our proposed methods, we carry the next move to identify how different layers of information take effects to enhance the quality of a summary in parameter tuning of α, β, γ and λ.

5.5 Parameter Tuning

Keeping other parameters fixed, we vary one parameter at a time to examine the changes of its performance from all 4 datasets. The first group of key parameters in our framework is α, β and γ where $\alpha + \beta + \gamma$=1. Every time we tune a parameter at a step of 0.1 and vary the other two for the best performance to achieve. Experimental results indicate the sentence-level relationship have stable but little impact on the summarization performance (illustrated in Fig. 2). The positive influence of documents and clusters are confirmed in Fig. 3 and Fig. 4 when $\beta \neq 0$ and $\gamma \neq 0$. Compared with document-level information, cluster-level information has a relatively weaker influence. Excessive use of higher level information impairs performance. Over smoothing from source texts might make the language models divergent from the original ones. We set α=0.3, β=0.5, γ=0.2 in our experiments.

Another key parameter in our framework is λ in Equation (8) to measure the tradeoff between visible and invisible cluster information. We gradually change λ from 0 to 1 at the step of 0.1 to examine the effect in Fig. 5. The combination of visible and invisible cluster outperforms the performance in isolation (λ=1 or 0). It is understandable that these two clustering metrics denote separate document organization methods and introduce different smoothing backgrounds. In general, a larger weight from visible cluster is preferable (λ=0.3).

Finally we examine the impact of document and cluster weights and the results are summarized in Table 6. From Table 6, we conclude that to distinguish the weight of documents and clusters is useful to measure sentence relationships because the usage of both weights brings prominent improve compared with $\pi(d)$=OFF and $\phi(C)$=OFF. We find that document weight by $\pi_{pr}(d)$ is much better than $\pi_{kl}(d)$, indicating that the web organization structure is helpful to find the centric documents within the corpus. The usage of $\pi_{kl}(d)$ has been proved in [12]. We also try different combinations of $\phi(C)$ for visible and invisible clusters. ϕ_{kl} means both clusters are weighed by KL-Divergence, and ϕ_{pr} means both clusters are weighed by PageRank score. ϕ_{kl+pr} means using KL-Divergence for visible clusters and using PageRank for invisible clusters, while ϕ_{pr+kl} means using PageRank score for visible clusters and using KL-Divergence for invisible clusters. We have an interesting finding that for visible clusters organized by anchor texts, the weights measured by PageRank seems to make more sense than using semantic coherence, and vice versa. Therefore, in general, the performance of ϕ_{pr+kl} is the most plausible weighting strategy.

Table 6. The impact of document weights and cluster weights, measured by KL-Divergence (kl), PageRank score (pr) and their different combinations

π \ ϕ	ON				OFF
	ϕ_{pr}	ϕ_{kl}	ϕ_{pr+kl}	ϕ_{kl+pr}	
π_{pr}	0.282	0.289	**0.291**	0.286	0.266
π_{kl}	0.268	0.271	0.273	0.265	0.254
OFF		0.242			0.237

6 Conclusions

In this paper we propose a Hierarchical Graph Summarization method, incorporating hybrid linkage information from multiple levels simultaneously into traditional graph summarization models. We utilize *sentence-to-sentence* relationship, *sentence-to-document* relationship and *document-to-cluster* relationship. We also investigate the web document structural information by explorations of **visible** and **invisible** document clusters, and the visible clusters earn heavier weights than invisible clusters (λ=0.3). Further more, we distinguish document/cluster by measuring their corresponding weights, calculating KL-Divergence and PageRank scores.

Abundant experiments are conducted on 4 real datasets, comparing 5 rival algorithms. Experimental results demonstrate the effectiveness of our proposed HGS. The benefits of visible and invisible clustering are also confirmed. Documents and clusters

should be distinguished by their significance. We also find that the semantic coherence for invisible clustering has not shown as promising effects as visible clustering does.

Acknowledgments. This work was partially supported by HGJ 2010 Grant 2011ZX01042-001-001 and NSFC with Grant No. 61073081. Xiaojun Wan was supported by NSFC with Grant No.61170166, and Rui Yan was supported by the MediaTek Fellowship.

References

1. Allan, J., Gupta, R., Khandelwal, V.: Temporal summaries of new topics. In: Proceedings of the 24th Annual International ACM SIGIR Conference, pp. 10–18 (2001)
2. Erkan, G., Radev, D.R.: Lexpagerank: Prestige in multi-document text summarization. In: Proceedings of EMNLP 2004, pp. 1–7 (2004)
3. Fukumoto, F., Suzuki, Y.: Extracting key paragraph based on topic and event detection: towards multi-document summarization. In: NAACL-ANLP 2000, pp. 31–39 (2000)
4. Kumaran, G., Allan, J.: Text classification and named entities for new event detection. In: Proceedings of the 27th Annual International ACM SIGIR Conference, pp. 297–304 (2004)
5. Li, L., Zhou, K., Xue, G.-R., Zha, H., Yu, Y.: Enhancing diversity, coverage and balance for summarization through structure learning. In: WWW 2009, pp. 71–80 (2009)
6. Lin, C.-Y., Hovy, E.: From single to multi-document summarization: a prototype system and its evaluation. In: Proceedings of ACL 2002, pp. 457–464 (2002)
7. Lin, C.-Y., Hovy, E.: Automatic evaluation of summaries using N-gram co-occurrence statistics. In: Proceedings of NAACL-HLT 2003, pp. 71–78 (2003)
8. Mei, Q., Zhai, C.: Generating Impact-Based Summaries for Scientific Literature. In: Proceedings of ACL 2008, pp. 816–824 (2008)
9. Mihalcea, R., Tarau, P.: A language independent algorithm for single and multiple document summarization. In: Proceedings of IJCNLP 2005, pp. 19–24 (2005)
10. Shen, C., Wang, D., Li, T.: Topic aspect analysis for multi-document summarization. In: Proceedings of CIKM 2010, pp. 1545–1548 (2010)
11. Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: Proceedings of AAAI 2008, pp. 855–860 (2008)
12. Wan, X.: An Exploration of document impact on graph-based multi-document summarization. In: Proceedings of EMNLP 2008, pp. 755–762 (2008)
13. Wan, X., Xiao, J.: Graph-based multi-modality learning for topic-focused multi-document summarization. In: Proceedings of IJCAI 2009, pp. 1586–1591 (2009)
14. Wang, D., Zhu, S., Li, T., Gong, Y.: Multi-document summarization using sentence-based topic models. In: Proceedings of ACL/AFNLP 2009 (Short Papers), pp. 297–300 (2009)
15. Wang, D., Li, T.: Document update summarization using incremental hierarchical clustering. In: Proceedings of CIKM 2010, pp. 279–288 (2010)
16. Yan, R., Wan, X., Otterbacher, J., Kong, L., Li, X., Zhang, Y.: Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In: Proceedings of the 34th Annual International ACM SIGIR Conference, pp. 745–754 (2011)
17. Yan, R., Nie, J.-Y., Li, X.: Summarize what you are interested in: an optimization framework for interactive personalized summarization. In: EMNLP 2011, pp. 1342–1351 (2011)
18. Zhai, C., Lafferty, J.D.: A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In: Proceedings of SIGIR 2001, pp. 334–342 (2001)

Mining Mobile Users' Activities Based on Search Query Text and Context

Bingyue Peng[1,⋆], Yujing Wang[2,*], and Jian-Tao Sun[3]

[1] Beihang University, Beijing 100191, China
[2] Key Laboratory of Machine Perception, Peking University, Beijing 100871, China
[3] Microsoft Research Asia, Beijing 100080, China

Abstract. Mobile search market is growing very fast. Mining mobile search activities is helpful for understanding user preference, interest and even regular patterns. In previous works, text information contained by either search queries or web pages visited by users is well studied to mine search activities. Since rich context information (e.g., time, location and other sensor inputs) is contained in the mobile search data, it has also been leveraged by researchers for mining user activities. However, the two types of information were used separately. In this paper, we propose a graphical model approach, namely the Text and Context-based User Activity Model (TCUAM), which mines user activity patterns by utilizing query text and context simultaneously. The model is developed based on Latent Dirichlet Allocation (LDA) by regarding users' activities as latent topics. In order to guide the activity mining process, we borrow some external knowledge of topic-word relationship to build a constrained TCUAM model. The experimental results indicate that the TCUAM model yields better results compared with text-only and context-only approaches. We also find that the constrained TCUAM model is more effective than the unconstrained TCUAM model.

Keywords: mobile user modeling, user's activity mining, Latent Dirichlet Allocation.

1 Introduction

With the prosperity of mobile market, more and more web search activities go to mobile devices. This raises the requirement of mining mobile search data, which is important for understanding user preferences, interests and activity patterns. Compared with web search from PC, mobile search data contains rich context information, e.g., time, location, surrounding business and other signals captured by sensors of mobile devices. Previous works of mining search activities focus on analyzing the content of search query, web pages, etc., with limited attentions of mining context information [3]. According to our analysis of mobile search log

⋆ This work was done when the first two authors conducted internship at Microsoft Research Asia.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 109–120, 2012.

Table 1. Three main types of user search activity data

User Behavior	Query Text	Search Context
Text-Dominated	**restaurant**	Time = 08:00∼09:00 Day = Monday SurroundingType = Amusement Equipment
Context-Dominated	facebook	**Time = 14:00∼ 15:00** **Day = Sunday** **SurroundingType = Baseball Clubs & Parks**
Both-Dependent	**Samsung focus price Amazon.com**	**Time = 15:00∼ 16:00** **Day = Saturday** **SurroundingType = Downtown**

data, we discover that both search query text and context information can help understand user activities. Table 1 gives examples of three major types of user search activities.

- **Text-Dominated** activities can be fully understood by query content, without considering context information. For example, query "restaurant" indicates that the user wants to find a restaurant.
- **Context-Dominated** activities can be explained by the context information. E.g., a user issues several queries with the following context: "Time = 14:00∼15:00", "Day = Sunday" and "SurroundingType = Baseball Clubs & Parks". We can infer that the user's activity may be related to "Playing Baseball".
- **Both-Dependent** activities require both text and context information to explain the user's activities. For instance, the user's context is "Time = 15:00∼16:00", "Day = Saturday", "SurroundingType = Downtown", and the query is "Samsung focus price Amazon.com". We can infer that the user's activity is likely to be "Shopping".

We can see that both text and context information can help understand the activity of mobile users. However, as far as we know, currently there are few approaches which can model user activities based on text and context information simultaneously.

In this paper, we propose a graphical model approach, namely the Text and Context-based User Activity Model (TCUAM) to mine user activity patterns using both query text and search context information. The TCUAM model is developed based on Latent Dirichlet Allocation (LDA), by regarding user activities as latent topics. As there are many noises in mobile log data, TCUAM has difficulty in discovering meaningful patterns. Therefore, we leverage human knowledge to help. We borrow external knowledge of topic-word relationship to build a constrained TCUAM model. The experiments on real mobile log indicates that the TCUAM model yields better results compared with text-only and context-only approaches. We also find that the constrained TCUAM model behaves more effectively than the unconstrained TCUAM model.

The rest of this paper is organized as follows. Section 2 briefly introduces the related work. The TCUAM model is defined in Section 3. We describe experiments and results in Section 4. Conclusions are given in Section 5.

2 Related Work

There are mainly two groups of research works which are related to ours. The first is about feature modeling used in mining user activity patterns. Understanding user intent from text information such as past queries and user profiles is a common technique for web search personalization. Sieg *et al.* [2] analyzed user profiles and assigned implicitly derived interest scores to existing concepts in a domain ontology for personalization. Noll *et al.* [4] implemented personalization using social bookmarking and tagging. Teevan *et al.* [5] utilized a personalization technique to leverage implicit information about the users' interests and activities, including previously issued queries, previously visited web pages and the documents a user has read or created. Besides text information mentioned above, context information is also adopted by researchers to mine user activity patterns. Arias *et al.* [6] found that it was beneficial to understand user intents and complete desired queries by context information such as time and location. Hattori *et al.* [7] improved the performance of query refinement by incorporating user context information. Church *et al.* [8] proposed a novel interface to support multi-dimensional and context-sensitive mobile search, combining context features such as location, time, and community preferences to offer better search experiences.

The other group of related works is about learning models. The models of utilizing context information can be divided into three stages. In the first stage, context information is manually processed in a certain domain, especially in a geographical system [9,10,11]. In the second stage, researchers begin to use traditional text learning model to tackle context learning problems. Algorithms like Bayesian Network, Hidden Markov Model (HMM), Support Vector Machine (SVM) and Conditional Random Field (CRF) have been adopted to model user behaviors [12,13]. In the third stage, unsupervised models are used for learning tasks of large-scale data. Topic model related approaches are adopted in this stage for user activity mining. E.g., Bao *et al.* [3] tried to model context information by an unsupervised approach based on latent dirichlet allocation (LDA) to discover users' activities.

3 Methodology

3.1 Data and Preprocessing

In this work, we mine user activities using mobile search log. The log contains a set of records $R = \{r_1, r_2 ... r_n\}$, where $r_i =< q_i, c_i >$. q_i is the query issued by the

user and c_i is the context when the search event happens. $c_i = \{< f_i, v_i > | 1 \leq i \leq N_p\}$ where $< f_i, v_i >$ is a feature-value pair, f_i stands for the feature name while v_i is the value of feature f_i. As we know, there are various noises in search log and it is difficult to obtain satisfactory results without data preprocessing. In traditional approaches, queries with low frequencies are usually regarded as noises and excluded from query log. In our work, we regard the queries which are not very related to context as noises. We introduce a scoring function to calculate the possibility of a query to be noise:

$$\xi(q) = -\frac{\sum_{i=1}^{N_p} p(< f_i, v_i >) log(p(< f_i, v_i >))}{|f_q - \tilde{f_{75\%}}|} \tag{1}$$

where $p(< f_i, v_i >)$ stands for the probability of feature f_i to take the value v_i. The smaller value $-\sum_{i=1}^{N_f} p(< f_i, v_i >) log(p(< f_i, v_i >))$ takes, the more irrelevant feature f_i is with the query text, thus the more likely query q will be a noisy query. f_q denotes the frequency of query q, $\tilde{f_{75\%}}$ stands for the average frequency of 75% queries whose values lie in the middle of all the queries. The larger value $|f_q - \tilde{f_{75\%}}|$ takes, the more extreme the query frequency is, thus the more likely the query is considered to be noise.

The smaller value $\xi(q)$ takes, the more likely that query q is considered as noise. In practice, an appropriate threshold is chosen to filter out noisy queries.

3.2 Text and Context-Based User Activity Model

In this section, we will introduce the Text and Context-based User Activity Model (TCUAM) for mining users' activities based on Latent Dirichlet Allocation (LDA). Bao *et al.* [3] has proposed an LDA-based approach to mine user activities using context information. However, context information itself can only explain part of user activities. Our model is designed to utilize text and context information simultaneously for user activity mining.

Given a set of records $R = \{r_1, r_2 ... r_n\}$, we split them into sessions according to time information. Each session contains data records within 30-minutes time span. Given a collection of M sessions $S = \{s_1, s_2 ..., s_M\}$, we assume that each session is generated by a collection of topics, which follow dirichlet distributions. Suppose there are totally K topics, $r_{m,n} =< q_{m,n}, c_{m,n} >$ denotes the n^{th} observation of record in the m^{th} session. $q_{m,n} = \{w_{m,n,1}, w_{m,n,2}, ...\}$ stands for the n^{th} observation of query text in the m^{th} session where $w_{m,n,i}$ is the i^{th} word of query $q_{m,n}$, $c_{m,n} = \{< f_i, v_i >\}$ stands for the n^{th} observation of context in the m^{th} session. $< f_i, v_i >$ represents a feature-value pair where f_i denotes the feature name and v_i denotes the value of feature f_i.

The process of generating text and context information for all the sessions can be expressed as follows. Firstly, draw a query word distribution φ_k for each topic k from dirichlet distribution with parameter β. Secondly, for each topic k and feature f, draw a feature-value pair distribution $\omega_{k \cdot f}$ from dirichlet

distribution with parameter τ. Thirdly, for each session s_m, draw a topic distribution θ_m and a feature distributions λ_m from dirichlet distribution with parameter α and γ respectively. Then for each session s_m, records are generated repeatedly based on the model. For each record $r_{m,n}$ in session s_m, we first choose a topic $z_{m,n}$ according to the topic distribution Multi(θ_m). Afterwards, query text and context information are generated respectively according to their distributions on topic $z_{m,n}$. Each word $w_{m,n,i}$ in the query text is generated directly from word distribution $Multi(\varphi_{z_{i,j}})$. For the context $c_{m,n} = \{< f_i, v_i >\}$, each feature f_i is generated from the feature distribution $Multi(\lambda_m)$ while the corresponding value v_i is generated from the feature-value pair distribution $Multi(\omega_{z_{i,j},f_i})$. Table 2 summarizes the generative process of TCUAM.

In practice, we take the whole query as a single word. That is, we use $q_{m,n} = w_{m,n}$, instead of $q_{m,n} = \{w_{m,n,1}, w_{m,n,2}, ...\}$. The graphical representation of the model is shown in Figure 1.

Table 2. Generative process of TCUAM

1. For each topic k
   Draw word distribution $\varphi_k \sim Dir(\beta)$
2. For each topic k and feature f
   Draw feature-value pair distribution $\omega_{k \cdot f} \sim Dir(\tau)$
3. For each session s_m
   (a) Draw topic distribution $\theta_m \sim Dir(\alpha)$
   (b) Draw feature distribution $\lambda_m \sim Dir(\gamma)$
   (c) For each record observation $r_{m,n}$ in session s_m
      (1) Choose a topic $z_{m,n} \sim$ Multi(θ_m)
      (2) Generate *query text* $q_{m,n}$:
         For each word in $q_{m,n}$
            Choose a word $w_{m,n,i} \sim$ Multi($\varphi_{z_{i,j}}$)
      (3) Generate *context information* $c_{m,n}$:
         For each feature value pair $< f_i, v_i >$ in $c_{m,n}$,
            (i) Choose a feature $f_i \sim$ Multi($\lambda_{m,z_{i,j}}$)
            (ii) Choose a feature value $v_i \sim$ Multi($\omega_{z_{i,j},f_i}$)

3.3 Inference of Model

To simplify the equations, we define the following symbols. The hyper-parameters in the TCUAM model are denoted as Θ, which include α, β, γ and τ. The observations are denoted by Γ, which consist of N_s sessions. $r_{m,n} =< q_{m,n}, c_{m,n} >$ denotes the n^{th} record in the m^{th} session. $q_{m,n} = w_{m,n}$ stands for the n^{th} observation of query text in the m^{th} session, and $c_{m,n} = \{< f_i, v_i > | 1 \leq i \leq N_p\}$ stands for the n^{th} observation of context in the m^{th} session. $< f_i, v_i >$ represents a feature-value pair where f_i denotes the feature name and v_i denotes the value of feature f_i. The parameters are represented by Δ, including θ, φ, λ and ω. The latent variables of topics are denoted by z. We define $\underline{\Phi} = \{\varphi_k\}_{k=1}^{K}$, $\underline{\Lambda} = \{\lambda_k\}_{k=1}^{K}$ and $\underline{\Omega} = \{\omega_p\}_{p=1}^{K \cdot F}$, where K is the total number of topics and F is the total

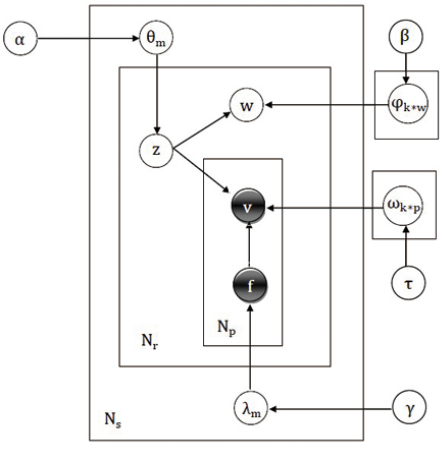


Fig. 1. Graphical Representation of TCUAM

number of features. Thus, given hyper-parameters, the joint distribution of all observations and hidden variables can be calculated as follows:

$$p(\Gamma, \Delta, z|\Theta) = \prod_{m=1}^{N_s} \prod_{n=1}^{N_r} p(w_{m,n}|\varphi_{z_{m,n}}) \prod_{j=1}^{N_p} p(v_p|\omega_{z_{m,n},f_j}) \\ \times p(f_p|\lambda_{m,z_{m,n}}) p(z_{m,n}|\theta_m) p(\theta_m|\alpha) p(\underline{\Phi}|\beta) p(\underline{\Lambda}|\gamma) p(\underline{\Omega}|\tau) \tag{2}$$

where N_s is the number of sessions, N_r is the number of records in each session, and N_p is the number of feature-value pairs in the context.

We obtain the joint probability for all observations by integrating over the parameters and latent variables:

$$p(\Gamma|\Theta) = \int\int\int\int p(\theta_m|\alpha) p(\underline{\Phi}|\beta) p(\underline{\Lambda}|\gamma) p(\underline{\Omega}|\tau) \\ \times \prod_{m=1}^{N_s} \prod_{n=1}^{N_r} \sum_{z_{m,n}} (w_{m,n}|\varphi_{z_{m,n}}) \prod_{j=1}^{N_p} p(v_p|\omega_{z_{m,n},f_p}) \\ \times p(f_p|\lambda_{m,z_{m,n}}) p(z_{m,n}|\theta_m) d\underline{\Phi} d\underline{\Lambda} d\underline{\Omega} d\theta_m \tag{3}$$

We use Gibbs sampling to get the approximate estimation of parameters. In Gibbs sampling, each record is assigned to a certain topic under the condition that other records have been labeled. We assume that *Text Information* and *Context Information* are generated by activity topics independently and obtain the following equation:

$$p(z_i = k|\boldsymbol{z}_{\neg i}, \boldsymbol{w}, \boldsymbol{c}) = p(z_i = k|\boldsymbol{z}_{\neg i}, \boldsymbol{w},) p(z_i = k|\boldsymbol{z}_{\neg i}, \boldsymbol{c}) \tag{4}$$

where $\boldsymbol{w}$ stands for the vector of words; $\boldsymbol{c}$ stands for the vector of feature-value pairs in the context; z_i represents the topic of the i^{th} record whereas $\boldsymbol{z}_{\neg i}$ denotes the vector of topics for all records after excluding the i^{th} record.

The two conditional probabilities on the right side of Eq.(4) can be calculated by [14]:

$$p(z_i = k|\boldsymbol{z}_{\neg i}, \boldsymbol{w}) = \frac{p(\boldsymbol{w}|\boldsymbol{z})}{p(\boldsymbol{w}|\boldsymbol{z}_{\neg i})} \cdot \frac{p(\boldsymbol{z})}{p(\boldsymbol{z}_{\neg \boldsymbol{i}})} \propto \frac{n^w_{k,\neg i} + \beta_w}{\Sigma^W_{w'=1} n^{w'}_k + \beta_w} \cdot \frac{n^k_{m,\neg i} + \alpha_k}{\Sigma^K_{k'=1} n^{k'}_m + \alpha_k} \tag{5}$$

$$\begin{aligned} p(z_i = k|\boldsymbol{z}_{\neg i}, \boldsymbol{c}) &\propto p(v_p|z_i = k, \boldsymbol{z}_{\neg i}, F, V_{\neg i}) p(z_i = k|\boldsymbol{z}_{\neg i}) \\ &= \prod_p^{N_p} \frac{n^{f_p,v_p}_{k,\neg i} + \omega_{v_p}}{\sum_{v'} n^{f_p,v'}_{k,\neg i} + \sum_{v' \epsilon V_{f_p}} \omega_{v'}} \cdot \frac{n^k_{m,\neg i} + \alpha_k}{\sum^K_{k'=1} n^{k'}_{m,\neg i} + \sum^K_{k'=1} \alpha_k} \end{aligned} \tag{6}$$

Where α_k denotes the hyper-parameter of dirichlet distribution for topic k, and β_w denotes the hyper-parameter of dirichlet distribution for word w. $i =< m, n >$ stands for the index of the n^{th} observation in the m^{th} session, $n^w_{k,\neg i}$ stands for the times of word w being observed with topic k after excluding the i^{th} record, $n^{f,v}_{k,\neg i}$ stands for the times of feature-value pair $< f, v >$ being observed with topic k after excluding the i^{th} record, and $n^k_{m,\neg i}$ stands for the times of topic k being observed in session m after excluding the i^{th} record.

After the convergence of Gibbs sampling iteration, each observation will be assigned a final topic label. Eventually, the parameters can be inferred as below:

$$p(w|z_k) = \varphi_{k,w} = \frac{n^w_k + \beta_w}{\Sigma^W_{w'=1} n^{w'}_k + \beta_w} \tag{7}$$

$$p(f_p, v_p|z_k) = p(v_p|z_k, f_p) p(f_p) \tag{8}$$

$$p(v_p|z_k, f_p) = \frac{n^{f_p,v_p}_k + \omega_{v_p}}{\sum_{v'} n^{f_p,v'}_k + n^{f_p,v_p}_k + \omega_{v' \epsilon V_{f_p}} \omega_{v'}} \tag{9}$$

$$p(f_p) = \frac{\sum^K_{k'=1} \sum_{v'} n^{f_p,v'}_{k'} + \lambda_{f_p}}{\sum_{f'} \sum^K_{k'=1} \sum_{v'} n^{f',v'}_{k'} + \sum_{f'} \lambda_{f'}} \tag{10}$$

3.4 Constrained TCUAM Model

In practice, the unconstrained TCUAM model is unable to achieve satisfactory results due to massive noises in mobile log. Therefore, we borrow some external knowledge about topic-word relationship to help. We leverage a set of websites, which are organized into a list of topics. For each topic, we can use search log to associate queries with websites using the follow-click information. Given a set of topics $K = \{k_1, k_2...\}$, assume that the set of websites for topic k_i is $url(k_i) = \{u_1, u_2...\}$. Suppose there is a set of queries $Q = \{q_1, q_2...\}$ and the

set of follow-click URLs for each query q_j is $fclick(q_j) = \{u_1, u_2...\}$. We split each query into word sequences. Thus, the relevant score of topic k_i and word w_j can be calculated by:

$$Score(k_i, w_j) = \sum_{w_j \epsilon q_t, u' \epsilon fclick(q_t), u' \epsilon url(k_i)} tfidf_{w_j} \tag{11}$$

We choose top 50 words with the highest relevance scores for each topic and map them to $\eta[50 \sim 1]$ linearly. For example, if *airline* is the third relevant word to topic *Travel*, $\eta(Travel, airline) = 48$. Thus, we obtain 50 representative words for each topic as external knowledge to guide the activity mining model. The core idea of using this knowledge is to increase the weight of an unlabeled word in Gibbs Sampling if it is known to be a representative word for a specific topic. The whole procedure of Gibbs sampling is displayed in Algorithm 1, where the variables are defined in Table 3.

Table 3. Variables in Algorithm 1

N_s	number of sessions to generate
N_r	number of records in a certain session
N_p	number of feature-value pairs in a certain record
η	relevance scores between words and topics
n_k^w	the times of word w being observed with topic k
n_k^p	the times of feature-value pair p being observed with topic k
n_k^f	the times of feature f being observed with topic k
n_m^k	the times of records being observed with topic k in document m
n_m	the times of records being observed in session m
n_k	the times of records being observed with topic k
n_p	the times of records being observed with feature-value pair p

4 Experiment

4.1 Data Set

In this paper, we carry out our experiments on real mobile logs from a commercial search engine. The data set consists of half a year's mobile logs in California State, USA. Table 4 shows the feature list of text and context information used in our experiment. For *period* feature, we define its values according to the time range of search activity. We remove the users whose query numbers are less than 50 in half a year time span. The preprocessing procedure described in section 3.1 is applied to clean the dataset. In our experiment, we set the threshold as 0.25.

The external knowledge of topic-word relationship for the constrained TCUAM model is illustrated in Table 5. We have 15 kinds of activity topics which mobile users are specially interested in. For each activity topic, 50 words are selected to be the external knowledge. Because of space limitation, only top 5 words for each activity topic are listed in the table.

Algorithm 1. Gibbs Sampling For Constrained TCUAM

zero all counter, $n_m^k, n_k^w, n_k^p, n_k^f, n_m, n_k, n_p$
for *each session* $s_m \epsilon [1, N_s]$ **do**
  for *each record* $r_{m,n} \epsilon [1, N_r]$ *in session* s_m **do**
    if *pre-word* $\tilde{k}$ *exist* **then**
      $r_{m,n}.topic = \tilde{k}$
    end
    else
      $r_{m,n}.topic = z_{m,n} \sim Mult(1/K)$
    end
    $k = r_{m,n}.topic$
    $\mathbf{S} = \eta(k, r_{m,n}.w)$
    Increase counter: $n_m^k + \mathbf{S}$, $n_m + \mathbf{S}$, $n_k^w + \mathbf{S}$, $n_k + \mathbf{S}$
    for *each feature-value pair* $p \epsilon [1, N_p]$ *in record* $r_{m,n}$ **do**
      Increase counter: $n_k^p + \mathbf{S}$, $n_k^f + \mathbf{S}$, $n_p + \mathbf{S}$
    end
  end
end

while *not converged* **do**
  for *each session* $s_m \epsilon [1, N_s]$ **do**
    for *each record* $r_{m,n} \epsilon [1, N_r]$ *in session* s_m **do**
      for the current record $r_{m,n}$ assigned to topic k: $\mathbf{S} = \eta(k, r_{m,n}.w)$
      Decrease counter: $n_m^k - \mathbf{S}$, $n_m - \mathbf{S}$, $n_k^w - \mathbf{S}$, $n_k - \mathbf{S}$
      $\diamond$ sample a new topic $k' \sim p(z_i = k' | \boldsymbol{z}_{\neg i}, \boldsymbol{w}, \boldsymbol{c})$, $\mathbf{S} = \eta(k', r_{m,n}.w)$
      Increase counter: $n_m^{k'} + \mathbf{S}$, $n_m + \mathbf{S}$, $n_{k'}^w + \mathbf{S}$, $n_{k'} + \mathbf{S}$
      for *each feature-value pair* $p \epsilon [1, N_p]$ *in record* $r_{m,n}$ **do**
        Increase counter: $n_{k'}^p + \mathbf{S}$, $n_{k'}^f + \mathbf{S}$, $n_p + \mathbf{S}$
      end
    end
  end
end

4.2 Experimental Setup

In the experiment, we evaluate four models which are described below.

- **TM** (Text-based Model) is the baseline of our experiment. The text-based model utilizes query text to build a LDA model for mining users' activities.
- **CM** (Context-based Model) is proposed by Bao *et al.* [3] to mine mobile users' activity patterns based on context information.
- **TCUAM** (Text and Context-based User Activity Model) is an unconstrained approach proposed in this paper to model users' activities by using text and context information collaboratively.
- **CTCUAM** (Constrained TCUAM) is a constrained model which uses external knowledge of topic-word relationship to guide the TCUAM model.

In order to get a fair comparison of the models above, we adopt the same session segmentation method for all the models. Records are segmented into sessions by time information. Each session contains the records within a time span of 30 minutes. In addition, the number of topics in all the models is set to be 200 experimentally.

Table 4. Feature Information

Data type	Feature	Feature-Value Range
Text Information	N/A	free query, pre-assigned query
Time Information	Date	05/01/2010, 05/02/2010, 05/03/2010... 12/31/2010
	Day	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
	WorkdayOrWeekend	Workday, Weekend
	Period	Early_Morning, Morning, Noon, Afternoon, Evening Night1, Night2
	Time	$01:00 \sim 02:00$, $02:00 \sim 03:00$ ... $23:00 \sim 00:00$
Location Information	GPS	Longitude and Latitude
	CityName	Glendale, San Diego, Rosemead, Los Angeles, Dublin...
	PostalCode	92128, 92880, 91361, 92107, 94804, 91737 ...
	SurroundingType	None, Colleges & Universities, Natural Gas Services...
	PlaceType	Home, Workplace, Other

4.3 Evaluation

The goal of our experiment is to examine whether users' activity patterns can be mined correctly from mobile logs. Unfortunately, it is difficult to identify automatically whether the result patterns make sense or not. Therefore, we examine the result topics produced by each model manually and assign each topic a score. The score is given according to the following rules:

- **5**: It is a perfect pattern and indicates user activity clearly.
- **4**: It is a good pattern and can give an overall sense of user activity.
- **3**: It is a reasonable pattern and gives some clues of the user activity.
- **2**: It is a bad pattern and includes many noises.
- **1**: It is a non-sense pattern and difficult to be understood.

The average score (AS) of result topics is calculated after each topic is assigned a score manually. In our experiment, we use AS as the metric to evaluate the performances of different models.

4.4 Results

Table 6 shows the results of different models, evaluated by the average score (AS). We can find out that the worst way to mine user's activity is the Context-based Model (CM), whose AS value is 1.995. It indicates that only context information is not enough to determine users' actual activities. The Text-based Model (TM) achieves 2.295 for AS value, which shows that the text information is more informative than the context information. By using text and context information simultaneously, the TCUAM model achieves 2.545 for AS value (improving 27.6% from Context-based Model and 10.9% from Text-based Model). Therefore, the text information and context information can be utilized collaboratively to benefit the activity mining approaches. The Constrained TCUAM model enhances the performance further by 11.8%, achieving an AS value of 2.845. Moreover, the knowledge used in the constrained model is easy to be collected. Thus, the constrained model can mine user activity topics more precisely without taking too much human efforts.

Table 5. Knowledge of Text Information

Shopping	Legal & Finance	Education	Travel
tanger	bank	middle	airline
store	union	school	hotel
deb	stock	university	airtran
outlet	credit	college	cathay
hollister	financial	institution	hyatt
Arts & Entertainment	Automotive & Vehicles	Business to Business	Home & Family
cinemark	toyota	store	badcock
cinema	ford	hollister	arien
theater	tire	suntrust	dyson
imax	honda	levi	home
krikorian	dodge	graco	rug
Government	Sports & Recreation	Health & Beauty	Food & Dining
library	yankee	hospital	restaurant
court	coach	doctor	steakhouse
ccap	dodger	medical	pizza
park	sporting	alzheimer	burger
civil	dunham	sentara	chili
Professionals & Services	Computers & Technology	Real Estate & Construction	
oregonian	sony	apartment	
kinko	garmin	region	
fimserve	safelink	hotel	
kroger	logmein	blum	
train	gps	comerica	

Table 6. Comparison of Models

Model	**5**	**4**	**3**	**2**	**1**	**Sum**	**AS**
TM	2	16	58	87	37	200	2.295
CM	3	11	31	92	63	200	1.995
TCUAM	7	29	57	80	27	200	2.545
CTCUAM	10	42	74	55	19	200	2.845

Table 7. Examples of activity topics

Text Information	IsRelevant	Text Information	IsRelevant
f stock	Yes	cocktail lounges	Yes
ewbc stock	Yes	sports bars	Yes
culos de caseras	Yes	night clubs	Yes
caty stock	Yes	restaurants	No
twitter search	No	carnivals	Yes
coh stock	Yes	amusement places	Yes
cellufun	No	fairgrounds	Yes
games	No	taverns	Yes
monster tits	No	norwalk amc	No
dis stock	Yes	barbecue restaurants	No
Context Information	IsRelevant	Context Information	IsRelevant
WorkdayOrWeekend=Workday	Yes	Period=Evening	Yes
PlaceType=Home	Yes	WorkdayOrWeekend=Weekend	Yes
Day=Wendensday	Yes	Day=Saturday	Yes
Period=Morning	Yes	Day=Tuesday	No
Period=Early_Morning	Yes	PlaceType=Other	Yes
Day=Tuesday	Yes	SurroundingType=Food & Dining	Yes
SurroundingType=None	No	Period=Afternoon	Yes
Time= 07 : 00 ∼ 08 : 00	Yes	Time= 17 : 00 ∼ 18 : 00	Yes
Time= 06 : 00 ∼ 07 : 00	Yes	Time= 19 : 00 ∼ 20 : 00	Yes
CityName=Glendale	No	Time= 18 : 00 ∼ 19 : 00	Yes

4.5 Case Study

To get a further understanding of the Constrained TCUAM model, we select some examples to demonstrate the results produced by the model. Table 7 shows

two topics discovered by the model. The "IsRelevant" column gives the human judgement that whether the text or context is relevant to the topic. It is easy to infer that the user's activity is *"searching for stock information at home in the workday morning"* for the left case and it is *"searching for amusement place after dinner outside in the weekend"* for the right case.

5 Conclusion

In this paper, we propose a text and context-based user activity model to mine user's activity patterns from mobile logs. In addition, we introduce a small amount of external knowledge about topic-word relationship to build a constrained TCUAM model. The experiments were carried out on real mobile logs. The experimental results have indicated that the TCUAM model can yield better results, compared with text-only and context-only approaches. We can also conclude from the results that the constrained TCUAM model performs more effectively than the unconstrained TCUAM model.

References

1. Wagner, M., Balke, W.-T., Hirschfeld, R., Kellerer, W.: A Roadmap to Advanced Personalization of Mobile Services. In: Proceedings of the DOA/ODBASE/CoopIS, Industry Program (2002)
2. Sieg, A., Mobasher, B., Burke, R.: Web Search Personalization with Ontological User Profiles. In: Proceedings of CIKM 2007 (2007)
3. Bao, T., Cao, H., Chen, E., Tian, J., Xiong, H.: An Unsupervised Approach to Modeling Personalized Contexts of Mobile Users. In: Proceedings of ICDM 2010 (2010)
4. Noll, M.G., Meinel, C.: Web Search Personalization Via Social Bookmarking and Tagging. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 367–380. Springer, Heidelberg (2007)
5. Teevan, J., Dumais, S.T., Horvitz, E.: Personalizing Search via Automated Analysis of Interests and Activities. In: Proceedings of SIGIR 2005 (2005)
6. Arias, M., Cantera, J.M.: Context-based Personalization for Mobile Web Search. In: Proceedings of VLDB 2008 (2008)
7. Hattori, S., Tezuka, T., Tanaka, K.: Context-Aware Query Refinement for Mobile Web Search. In Proceedings of SAINT-W 2007 (2007)
8. Church, K., Smyth, B.: Who, What, Where & When: A New Approach to Mobile Search. In: Proceedings of IUI 2008 (2008)
9. Gregory, D.A., Atkeson, C.G., Hong, J., Long, S.: Kooper,R, Pinkerton, R.: Cyberguide: A Mobile Context-Aware Tour Guide. Wireless Networks (1997)
10. Ozturk, P., Aamodt, A.: Towards a Model of Context for Case-based Diagnostic Problem Solving. In: Proceedings of CONTEXT 1997 (1997)
11. Schilit, B., Adams, N., Want, R.: Context-Aware Computing Applications. In: Proceedings of the Workshop on Mobile Computing Systems and Applications (1994)
12. Liao, L., Patterson, D. J., Fox, D., Kautz, H.: Building Personal Maps from GPS Data. In: Proceedings of IJCAI Workshop on Modeling Others from Observation (2005)
13. Darnell, M.H.H., Moore, J., Essa, I.A.: Exploiting Human Actions and Object Context for Recognition Tasks. In: Proceedings of ICCV 1999 (1999)
14. Heinrich, G.: Parameter Estimation for Text Analysis. Technical Report (2004)

Spread of Information in a Social Network Using Influential Nodes

Arpan Chaudhury, Partha Basuchowdhuri*, and Subhashis Majumder

Heritage Institute of Technology,
Department of Computer Science and Engineering,
Chowbaga Road, Anandapur, Kolkata 700107, WB, India
arpanchaudhury@gmail.com,
{parthabasu.chowdhuri,subhashis.majumder}@heritageit.edu

Abstract. Viral marketing works with a social network as its backbone, where social interactions help spreading a message from one person to another. In social networks, a node with a higher degree can reach larger number of nodes in a single hop, and hence can be considered to be more influential than a node with lesser degree. For viral marketing with limited resources, initially the seller can focus on marketing the product to a certain influential group of individuals, here mentioned as *core*. If k persons are targeted for initial marketing, then the objective is to find the initial set of k active nodes, which will facilitate the spread most efficiently. We did a degree based scaling in graphs for making the edge weights suitable for degree based spreading. Then we detect the *core* from the maximum spanning tree (MST) of the graph by finding the top k influential nodes and the paths in MST that joins them. The paths within the *core* depict the key interaction sequences that will trigger the spread within the network. Experimental results show that the set of k influential nodes found by our *core* finding method spreads information faster than the greedy k-center method for the same k value.

Keywords: spread of information, social network analysis, maximum spanning tree, k-center problem.

1 Introduction

1.1 Motivation

A social network is a graph that represents relationships and interactions between a group of individuals. It acts as a medium through which information, innovations and influence spread among its members. An idea forked up from a community or an individual can either disappear with passage of time or influence a significant number of members in the network. For industry-based market analysts, the most interesting feature about a social network is that when people start recommending a new product to their friends, the product gains popularity

* Corresponding author.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 121–132, 2012.

very quickly. The strategy of marketing a product by targeting a small number of individuals, which trigger *brand awareness* [1] among all the members of the network through self-replicating viral diffusion of messages, is known as *viral marketing* [2,3,4]. Viral marketing can be much more cost effective than traditional methods since it employs the customers themselves to accomplish most of the promotional effort. Further, as people trust recommendations from their friends more than the manufacturing company itself, viral marketing pays off handsomely. The only challenge we face while utilizing *word-of-mouth* [5,6] advertisement is that we need to pick out a set of customers that maximizes the information flow within a network. For example, suppose we have a social network where the extent to which individuals influence one another is known and we want to endorse a new product in the network. We have a limited budget which is sufficient to convince at most k members to adopt the product. These k members of the network having the information are referred to as the *initial active node set* S and the rest of the nodes, who do not have the information yet, are called *inactive nodes*. The influence spreads from one node to another with time and the *active node set* grows (similarly *inactive node set* decreases) until further spread is not possible.

The problem mentioned above is known as the *influence maximization* problem, which was first introduced by Kempe et al. [7,8] as a discrete optimization problem. In this paper, we put forward an efficient heuristic which improves existing algorithms for influence maximization from two complementary directions. One is to propose a new heuristic that spreads the influence to maximum number of nodes within minimum amount of time and the second is to improve the greedy algorithm to further reduce its run-time. In this section we provide a brief introduction to the problem that we have solved and we also discuss some of the important works related to spread of information that is relevant to our work. In the next section, we have discussed our approach for an efficient spread of information in a network and describe our algorithm elaborately. In the third section, we have discussed about the experimental results describing the performance of our algorithm compared to pre-existing algorithms, we conclude by highlighting our contributions in the section thereafter.

1.2 Literature Review

It is a widely accepted fact that with proper choice of *influential mediators* [9] information can circulate within the network in minimum time. The optimization problem of finding such influential nodes in a social network was first introduced by Domingos and Richardson [2,3]. Motivated by its application in viral marketing, Kempe et al. [7,8] studied the *influence maximization* problem, considering two fundamental propagation models - *linear threshold model* (LT) and *independent cascade model* (IC). They showed that *influence maximization* problem is NP-hard and a simple greedy algorithm of successively selecting influential mediators approximates the optimum solution within a factor of $(1 - \frac{1}{e})$.

Later, Even-Dar and Shapira extended the study of *spread maximization set* problem where the underlying social network behaves like the *voter model*. In

their paper [10], they proposed an algorithm that gives an exact solution to the abovementioned problem, when all nodes have the same cost (cost of introducing a person to a new technology/product), and also provided a *fully polynomial time approximation scheme* for the more general case in which different nodes may have different costs. Kimura and Saito proposed *shortest path based influence cascade models* [11] and provided efficient algorithms to compute spread of influence under these models. Recently, Kimura et al. [12] proposed an alternative method for finding a good approximate solution to the *influence maximization* problem on the basis of *bond percolation* and graph theory. Using large-scale real networks including blog networks they experimentally demonstrated that the method proposed by them is much more efficient than the conventional methods.

Another well-studied problem that we refer to in this paper is the k-center problem [13,14,15,16]. It is defined as a facility location problem where the objective is to find appropriate locations for the facilities such that maximum distance from any client to its nearest facility is minimized. A close observation on k-center problem shows that it is very much similar to the *influence maximization* problem, as in both the cases we try to find a set of nodes which facilitate the service or information spread. In our paper, we show that selecting influentials based on their degrees can produce even better result than existing algorithms and that too in much less time. Knowledge of the related works mentioned in this section gives us an overview of spread of information. However, the algorithm that we have presented in this paper, approaches the problem differently from the existing models.

2 Maximizing Influence Spread

2.1 Problem Definition

Assuming each member of a social graph spreads information to its neighbors with probability 1, we aim at solving the following problem,

Problem 1: *Given a social network graph $G = (V, E)$ with a weight vector W indicating the extent to which individuals influence one another, find a set S of influential mediators of cardinality at most k, such that the objective function is defined as,*

$$r = \max_{v \in V} d(v, S) \tag{1}$$

and is minimized where,

$$d(v, S) = \min_{i=1}^{k} d(v, s_i) \tag{2}$$

and $d(a, b)$ is the shortest distance between nodes a and b.

2.2 Our Approach

In this paper, our primary objective is to find an initial set of *active* nodes in a social graph which maximizes propagation of a new innovation within the

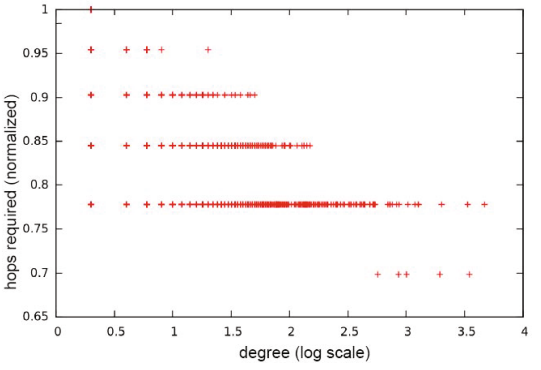


Fig. 1. Correlation between the degree of nodes in a network and the time it takes to spread the information throughout the network (AS relationship data, with 6474 nodes and 13895 edges), if the spread is simulated assuming that initial active node set consists of that node only

network in minimum time. For example, adoption of new drug within the medical profession, use of cell phone among school students, etc. For a social graph $G = (V, E)$, we consider the problem of finding a subgraph $G_c = (V_c, E_c)$, where $V_c \subseteq V$ and $E_c \subseteq E$, through which maximum information flow is likely to happen. We define this region as *core* of the graph. Initially we scale the weight of an edge $e \in E$, by the average of the degrees of the two nodes connected to e. Based on the notion that greater the degree of a node, higher the influence it imparts on the social network due to its ability to reach greater number of nodes (refer to Fig 1). It is desirable to use the edges that are incident on nodes having higher degree. Hence we use this average degree value as a multiplicative factor to the existing edge weights. In case of unweighted graphs, initial edge weights for all edges are taken to be 1 and for weighted graphs, some existing edge weights are assumed to be provided. These initial edge weights have been denoted as $weight_{old}(e_{ij})$ in equation 3. The basic idea is to include high-degree nodes within the *initial active node set*, so that reachability to other nodes within one hop is maximized from the very first step of the spread. Hence, it is also important to track the interactions or the edges between nodes with high influence. To track such edges, we define an objective function to scale the weight of each edge of the graph with the average degree of the nodes connected by that edge.

Definition 1: *Given a social network graph $G = (V, E)$, where $\forall e_{ij} \in E$, e_{ij} denotes an unordered pair of nodes (v_i, v_j), and $v_i, v_j \in V$. We denote the existing weight of e_{ij} by $weight_{old}(e_{ij})$, and then we define the revised weight of an edge to be*

$$weight(e_{ij}) = weight_{old}(e_{ij}) \times \frac{degree(v_i) + degree(v_j)}{2} \tag{3}$$

After scaling the edge weights of the graph, we aim to find the maximum cost spanning tree of the weighted graph. This problem is same as finding a minimum cost spanning tree of an isomorphic graph G_{iso} that has a one to one mapping for all the nodes and edges in G, where the edge weights are of same absolute value but with negative signs. Prim's algorithm for finding minimum cost spanning tree is quite popular and is used on G_{iso}. This minimum cost spanning tree generated from G_{iso} gives us the tree, which can be re-mapped to the labels in G and hence the maximum cost spanning tree of G can be found.

The above representation gives us edge weights based on the influence of the nodes. The function used for defining weights of the edges was motivated by the fact that finding a maximum spanning tree from the weighted graph would give us the path by which a node is connected to its neighbor with highest degree. Hence the maximum spanning tree would generate the path that is most likely to be followed if the influence starts to spread from the nodes with highest degree.

Definition 2: *The maximum spanning tree of a graph G is a connected subgraph $G_T = (V, E_T)$, where $E_T \subseteq E$ and $\forall e_{T_i} \in E_T$,*

$$\sum_{i=1}^{|E_T|} weight(e_{T_i}) \geq \sum_{i=1}^{|E_k|} weight(e_{k_i}) \tag{4}$$

for any E_k, where $E_k \subseteq E$ and $\forall e_{k_i} \in E_k$, forming a spanning tree. The edge weights here essentially denote the strength of interactions between adjacent nodes. So, we essentially scale the existing weight based on the topological structure of the graph and include the significance of the degree of vertices within the edge weights. Attributed graph may have different edge weights for the same edge based on different features. As for example, in a zonal call graph of a cellular service provider, interactions between any two users can be judged by the number of calls or the number of SMSs or some other mode of interactions between them. Strength of such interactions can be judged by the number of calls/week or number of SMSs/week basis. As long as a single composite edge weight based on some objective function can be deduced from the edge weights for each feature, we can also use this method for attributed graphs. However, the ways of finding such composite edge weights, remains out of the scope of this paper. If the edge weights are only determined by some apriori information, the effects of the graph topological structure can be ignored. Hence, in order to take into account both the externally collected information as well as the knowledge of the graph topological structure, we scale the initial edge weights to convert them into new edge weights.

It should be noted that in case of disconnected graphs, if we try to get the maximum spanning tree, not all the nodes will be included in the tree. So, it would only be meaningful to pick the largest connected component of the original graph as the graph, where the spread of information is observed and find the maximum spanning tree from it. Here, G is considered as the largest

connected component of the original graph. Usually for social graphs, largest connected components consist of around 95% (or more) of the nodes in the graph. After selecting the largest connected component and extracting the maximum spanning tree from it, we will have a unique path between any pair of nodes.

The maximum spanning tree, at this stage, consists of a subset of the edge set E using which maximum amount of information flows, but it still consists of all the nodes as in V. For social graphs with fairly large number of actors(nodes), influencing all the nodes immediately requires huge marketing expenses. So, our objective is to select a few nodes with topmost degrees of the network, target to market the product to those influential nodes so that they could spread the influence in as less number of steps as possible. Finding the *core* of the graph provides us with a trade-off between the budget and the time of spread. It does not require the product to be marketed to everyone i.e, the nodes with lower influence can be ignored. Hence, this model can work with a restricted budget. But at the same time instead of influencing everyone in one step, it takes more number of steps to reach all the nodes in the graph. The number of steps to reach all or the majority of the nodes needs to be optimized by suitably choosing the top k influential individuals. We follow some rudimentary graph coarsening techniques to reduce the number of nodes so that we can follow the behavior of the *cores* with various influence limits and become aware of their structures. In order to coarsen the graph, we pick a certain degree threshold based on the point where the degree distribution plot of the nodes in V has the least slope. If the degree threshold is denoted by d_{th} then the final graph that represents a *core* for the threshold d_{th} is denoted by $G_x = (V_x, E_x)$ where $\forall v_{x_i} \in V_x$, $degree(v_{x_i}) \geq d_{th}$. Higher the value of d_{th}, lower will be the cardinality of V_x.

In some cases, where coarsening the graph results in formation of a disconnected *core*, we introduce *bridge* nodes to join the components. Addition of *bridge* nodes in influential node set enhances the chance of knowledge propagation between influentials and hence different communities and thereby increases the spread within the network [17]. Influential nodes may exist in disjoint clusters in different parts of the network. In that case, these *bridge* nodes work as brokers of information from one of those clusters to another. For example, in ancient or medieval age epidemic break-outs stayed within a geographic location as communication between geographic regions was restricted, therefore restricting the *brokers*. But recently, during spread of swine flu, which generated from Mexico, some individuals (here *brokers*) helped its spread to even geographically distant locations like eastern Asia. Intuitively, these *brokers* should have higher edge betweenness values than other nodes in the network.

Note that, in this model we are assuming that a node, who gets activated at time-stamp t, always transmits the information to its neighbors and the inactive neighbors accept the information to become activated at time-stamp $t+1$. If acceptance of information by the neighbors becomes probabilistic, then the model becomes probabilistic too. We plan to follow-up this work with a probabilistic model of influence spread using the *core* as a seed for the spread.

2.3 Detecting the *core*

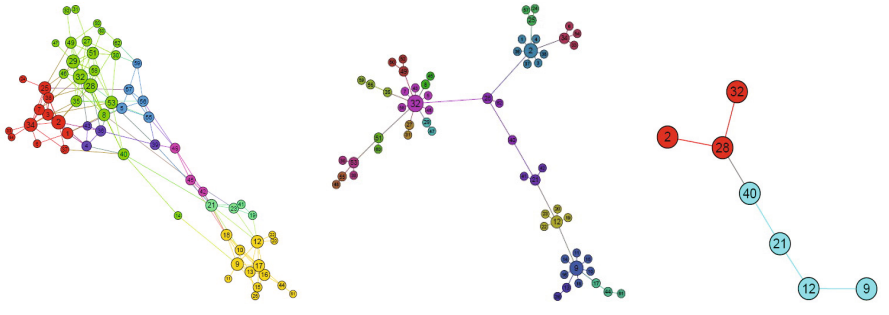
In this section, we explain the algorithm for finding the *core*, as defined in the previous section. Given the graph G and modified weight vector W we find the core G_c using this algorithm. In line 1, we use Prim's algorithm to find the maximum spanning tree and store it as G_T. The vertex and edge set of G_c are initialized in line 3. In lines 4-13, we get the nodes with degree value higher than degree threshold(d_{th}) and connect the maximum spanning tree edges between

Algorithm 1. Detecting the *core* of a graph

Input : $G(V,E)$ the social network, $weight(e_{ij}) \in W, \forall e_{ij} \in E$
Output: The *core* $G_c = (V_c, E_c)$

1 $G_T(V, E_T) \leftarrow MSTPrim(G, weight_{ij})$
2 $//MSTPrim(G, weight_{ij})$ finds the maximum spanning tree of G using Prim's algorithm
3 $V_c \leftarrow \emptyset,\ E_c \leftarrow \emptyset$
4 **for** *each vertex* $v \in V$ **do**
5 **if** $degree(v) \geq d_{th}$ **then**
6 then $V_c \leftarrow V_c \cup \{v\}$
7 **end**
8 **end**
9 **for** *each vertex* $e_{ij} \in E_T$ **do**
10 **if** $u, v \in V_c$ **then**
11 then $E_c \leftarrow E_c \cup \{e_{ij}\}$
12 **end**
13 **end**
14 // In MST, path between any pair of nodes v_i, v_j gives us a tree, defined as $G_{path(i,j)} = (V_{path(i,j)}, E_{path(i,j)})$
15 $G_{cc} \leftarrow DepthFirstSearch(G_c)$
16 // If G_c is disconnected, let $G_{cc} = \{G_{cc1} \cup G_{cc2} \cup G_{cc3} \cup ... \cup G_{ccp}\}$, where $G_{cci} = (V_{cci}, E_{cci})$
17 // Given $i, j,\ where\ i < j, v_i \in V_{cci}, v_j \in V_{ccj},\ G_{cci}, G_{ccj} \in G_{cc}$
18 **repeat**
19 **foreach** *pair* $G_{cci}, G_{ccj} \in G_{cc}$ **do**
20 **if** $\exists k, V_{cck} \subseteq V_{cc}\ and\ \exists v_l \ni v_l \in V_{cck}, V_{path(i,j)}, v_l \notin V_{cci}, V_{ccj}$ **then**
21 $V_{cci} \leftarrow V_{cci} \cup V_{ccj} \cup V_{path(i,j)} \cup V_{cck1} \cup V_{cck2} \cup ... \cup V_{cckr}$
22 $E_{cci} \leftarrow E_{cci} \cup E_{ccj} \cup E_{path(i,j)} \cup E_{cck1} \cup E_{cck2} \cup ... \cup E_{cckr}$
23 $V_{cc} \leftarrow V_{cc} - V_{ccj} - V_{cck1} - V_{cck2} - ... - V_{cckr}$
24 $E_{cc} \leftarrow E_{cc} - E_{ccj} - E_{cck1} - E_{cck2} - ... - E_{cckr}$
25 **else**
26 $V_{cci} \leftarrow V_{cci} \cup V_{ccj} \cup V_{path(i,j)}$
27 $E_{cci} \leftarrow E_{cci} \cup E_{ccj} \cup E_{path(i,j)}$
28 $V_{cc} \leftarrow V_{cc} - V_{ccj}$
29 $E_{cc} \leftarrow E_{cc} - E_{ccj}$
30 **end**
31 **end**
32 **until** G_{cc} *consists of only* G_{cc1};

them. In this way, we get the *influential* nodes but the *broker* nodes are yet to be accounted for. Also, such that the *core* at this stage may be disconnected. So we run depth first search (DFS) on G_c and store the components in G_{cc}. In essence, G_c and G_{cc} are same. If G_c is disconnected, then G_{cc} would be union of multiple disjoint graph components. In lines 18-32, we keep merging the components until one single connected component is produced. In this process, for all pair of components we select any node from each of them and try to find the path between them from E_T. While adding, any node external to V_c might be added. Note that the path between the two components may go through other components. In those cases, all these components are merged into one. The process continues until G_c becomes one connected component. Due to the use of maximum spanning tree for its generation, the final *core* turns out to be a tree (refer to Fig 2 and Fig 3). The run-time of the algorithm is dominated by the step where Prim's algorithm (using binary heap) is being called i.e. $O(|E|log|V|)$.



(a) Dolphin network with 62 vertices and 159 edges

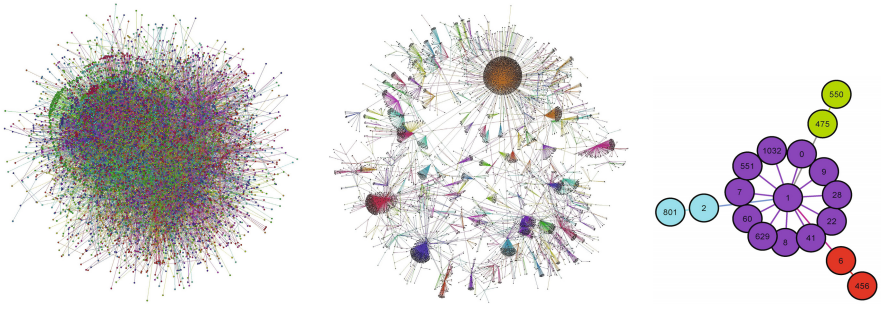
(b) Maximum spanning tree of Dolphin network

(c) *Core* of Dolphin network, with $k = 7$

Fig. 2. Diagrams of the original dolphin interaction network, its maximum spanning tree and *core* with k=7. The coloring of nodes depicts communities within the network and its purpose is to generate a better visualization.

3 Experimental Results

We have tested the quality of *influence maximization set* generated by our method on a number of social networks which have been studied by several authors [18,17,19]. We executed our algorithm for *core* finding and spread of information on a desktop PC with 2.0 GHz Intel core duo processor, 3 GB RAM and LINUX Ubuntu 10.10 OS. We also compared the accuracy of our heuristic with a popular spread maximization method. For graph visualization we used Gephi [20], an open source software for exploring and manipulating networks. All the programs developed for experiment purpose, have been written in C++ and was compiled with GNU g++ 4.6.0 compiler.



(a) AS relationship network with 6474 nodes and 13895 edges

(b) Maximum spanning tree of AS relationship network

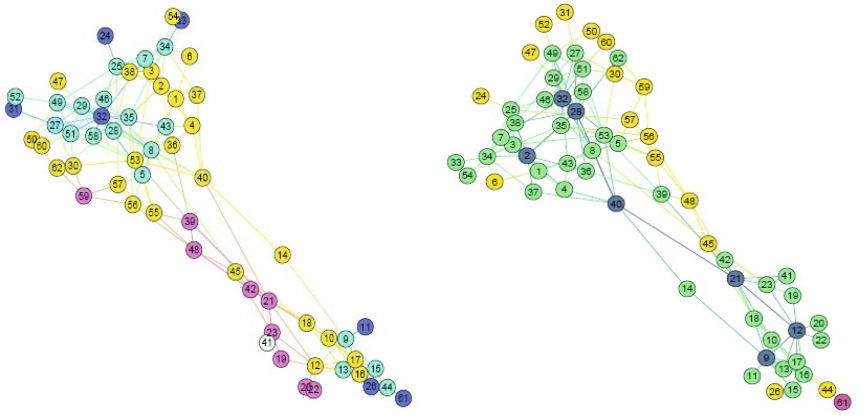
(c) *Core* of AS relationship network, with $k = 18$

Fig. 3. Diagrams of the smaller AS relationship network, its maximum spanning tree and *core* with k=18 (d_{th}=50). The coloring of nodes depicts communities within the network and its purpose is to generate a better visualization.

Table 1. *Core* finding method performs better than greedy k-center overall. Higher the number of k, better the performance of *core* finding method over k-center method.

Data	Number of k	Hops to spread information to 99% of the network nodes	
		greedy k-center	*core* finding
Zachary's Karate Club	3	2	2
Dolphin Network	7	4	3
ArXiv GrQc collaboration	9	7	7
	6	7	7
AS relationship network (small)	18	4	3
	10	4	3
	3	4	4
AS relationship network (large)	30	4	3
	12	4	4
	6	4	4

We have performed the experiments on a total of five different social network datasets of different size. The first one, is Zachary's karate club data, a social network of friendships between 34 members of a karate club at a US university in the 1970s [18]. The second one is an undirected social network of frequent associations between 62 dolphins in a community living off a coastal region of New Zealand. The third dataset, GR-QC (General Relativity and Quantum Cosmology) collaboration network, is from the e-print arXiv and covers co-author relationships between scientists, who submitted their papers to the General Relativity and Quantum Cosmology category between 1993 to 2003. It consists of 5242 nodes and 28980 edges. The other two datasets are, AS-relationship

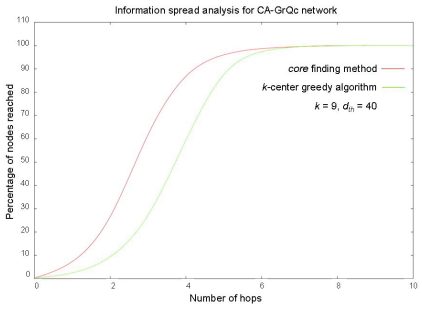


(a) Spread using k-center (b) Spread using *core*

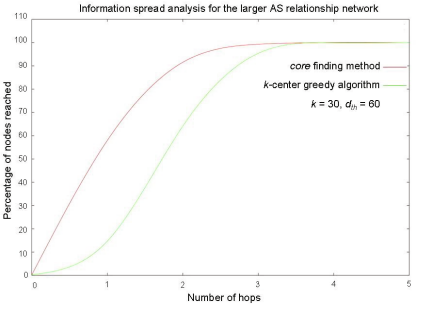
Fig. 4. Visual representation of the spread of information in the dolphin interaction network. Initial active node set is denoted by blue colored nodes. The nodes getting the information at first hop are colored green/cyan, the nodes at second hop are colored yellow, the nodes at third hop are colored pink and the nodes getting the information at the fourth and final step are colored white.

datasets from CAIDA website [19]. AS-relationships are important for routing policies and has implications on network robustness, traffic engineering, macroscopic topology measurement strategies. We use two AS-relationship datasets of different size, to observe how our algorithm performs on network of similar structure but of different size. One of these two datasets has 6474 nodes and 13895 edges and the other has 16301 nodes and 32955 edges.

We have compared our method with k-center problem, which is also a facility location problem and distributes the facilities within the network in such a way so that the maximum distance from all the nodes to its nearest facility is minimized. This is essentially another way to model the spread where the facility locations could be selected for initiating the spread. For all the instances of our experiments, we have seen that the *core* finding method works faster or at least as fast as the greedy solution for the k-center problem. In *core* finding, value of k is determined by d_{th}. From Table 1, it seems that with higher value of k, *core* finding performs better than the greedy solution for the k-center problem. Comparative performance between these two methods for some k and d_{th} combinations using all five datasets have been shown in Table 1, Fig 4 and Fig 5. An important observation from the experimental results is that, even if we increase the value of k, number of steps to reach the information to 99% of the nodes in the network does not necessarily reduce. Say, due to budget constraints, we want to choose k to be 7. Based on input value of k, say, by using the algorithm, we get the number of hops to reach every node in a network from its *core* to be 4. From another observation, we may also get to see that in that same network



(a) Rate of spread in ArXiv GrQc collaboration network



(b) Rate of spread in AS relationship network (large)

Fig. 5. Comparative study between the rates of spread of information using k-center method and *core* finding method, in CA-GrQc and AS-relationship(large) datasets

we can achieve the spread to all nodes with 4 hops for k=5 too. In that case, we need to find the lowest value of k for which the number of hops still remain the same as in case of the input k value. In such a situation, remaining within the budget constraints, no faster spread will be possible but it will be possible that not all the budget will be used up for initial marketing or creating the *initial active set* of nodes. Hence, a lower number of nodes may also be able to spread the information in same time. We want to extend our work by efficiently finding the lowest k values for all set of hops.

4 Conclusion

In this paper, we have presented an efficient method for spread of information by selecting the influential nodes based on degree. We have proposed a technique of scaling existing edge weights based on the degree of the two nodes on which the edge is incident. Using the new scaled edge weights, we have proposed a method to find an important set of nodes from the network and have named it as *core*. We have selected this *core* as the seed or the initial set of *active* nodes for the spread of information and have shown that the spread using the *core* works faster than greedy k-center method.

References

1. Kiss, C., Bichler, M.: Identification of influencers - measuring influence in customer networks. Decision Support Systems 46, 233–253 (2008)
2. Domingos, P., Richardson, M.: Mining the network value of customers. In: KDD, pp. 57–66 (2001)
3. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: KDD, pp. 61–70 (2002)

4. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. TWEB 1(1) (2007)
5. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters 12, 211–223 (2001)
6. Goldenberg, J., Libai, B., Muller, E.: Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. Academy of Marketing Science Review, 118 (2001)
7. Kempe, D., Kleinberg, J.M., Tardos, É.: Maximizing the spread of influence through a social network. In: KDD, pp. 137–146 (2003)
8. Kempe, D., Kleinberg, J.M., Tardos, É.: Influential Nodes in a Diffusion Model for Social Networks. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1127–1138. Springer, Heidelberg (2005)
9. Li, C.-T., Lin, S.-D., Shan, M.-K.: Finding influential mediators in social networks. In: WWW (Companion Volume), pp. 75–76 (2011)
10. Even-Dar, E., Shapira, A.: A Note on Maximizing the Spread of Influence in Social Networks. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 281–286. Springer, Heidelberg (2007)
11. Kimura, M., Saito, K.: Tractable Models for Information Diffusion in Social Networks. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 259–271. Springer, Heidelberg (2006)
12. Kimura, M., Saito, K., Nakano, R., Motoda, H.: Extracting influential nodes on a social network for information diffusion. Data Min. Knowl. Discov. 20, 70–97 (2010)
13. Hochbaum, D.S., Shmoys, D.B.: A best possible heuristic for the k-center problem. Mathematics of Operations Research 10, 180–184 (1985)
14. Gonzalez, T.F.: Clustering to minimize the maximum intercluster distance. Theor. Comput. Sci. 38, 293–306 (1985)
15. Mihelic, J., Robic, B.: Solving the k-center problem efficiently with a dominating set algorithm. CIT 13, 225–234 (2005)
16. Berger-Wolf, T.Y., Hart, W.E., Saia, J.: Discrete sensor placement problems in distribution networks. Mathematical and Computer Modelling 42, 1385–1396 (2005)
17. Lusseau, D., Newman, M.E.J.: Identifying the role that individual animals play in their social network. Proc. R. Soc. London B 271, S477 (2004)
18. Zachary, W.W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33, 452–473 (1977)
19. Dimitropoulos, X., Hyun, Y., Krioukov, D., Fomenkov, M., Riley, G., Huffaker, B.: As relationships: Inference and validation. Comput. Commun. Rev. (2007)
20. Bastian, M., Heymann, S., Jacomy, M.: Gephi: An open source software for exploring and manipulating networks. In: International AAAI Conference on Weblogs and Social Media (2009)

Discovering Coverage Patterns for Banner Advertisement Placement

P. Gowtham Srinivas, P. Krishna Reddy, S. Bhargav,
R. Uday Kiran, and D. Satheesh Kumar

International Institute of Information Technology Hyderabad, India
{gowtham.srinivas,bhargav.spg08,
uday_rage,satheesh.kumar}@research.iiit.ac.in,
pkreddy@iiit.ac.in

Abstract. We propose a model of coverage patterns and a methodology to extract coverage patterns from transactional databases. We have discussed how the coverage patterns are useful by considering the problem of banner advertisements placement in e-commerce web sites. Normally, advertiser expects that the banner advertisement should be displayed to a certain percentage of web site visitors. On the other hand, to generate more revenue for a given web site, the publisher has to meet the coverage demands of several advertisers by providing appropriate sets of web pages. Given web pages of a web site, a coverage pattern is a set of pages visited by a certain percentage of visitors. The coverage patterns discovered from click-stream data could help the publisher in meeting the demands of several advertisers. The efficiency and advantages of the proposed approach is shown by conducting experiments on real world click-stream data sets.

Keywords: Click stream mining, online advertising, internet monetization, computational advertising, graphical ads delivery.

1 Introduction

We have proposed a model of data mining pattern, called, "coverage patterns" and a methodology to discover coverage patterns from transactional databases. Given a set of data items, a coverage pattern is a set of non-overlapping data items covered by a certain percentage of transactions. An Apriori-like algorithm [1] called *CMine* is proposed for mining coverage patterns.

In the literature, the notion of coverage is being used for solving the set cover problem [2] in set theory and node cover problem [3] in graphs respectively. In [4], the notion of coverage and overlap is used to examine the creation of a tag cloud for exploring and understanding a set of objects. In [5], the notion of coverage and overlap is used to solve the problem of topical query decomposition. In this paper, we have proposed a different kind of knowledge patterns. The proposed patterns can be employed in improving the performance of several applications such as banner advertisements.

The research in this paper is motivated with the problem of banner advertisement placement. The background and problem description is as follows.

Banner advertising is one of the dominant modes of online advertising, in addition to the contextual and sponsored search advertising. A banner advertisement is described as

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 133–144, 2012.

a hypertext link that is associated with a box containing graphics which is redirected to a particular web page when a user clicks on the banner [6]. The following three entities are involved in banner advertising: advertiser, publisher and visitor. An advertiser is interested in endorsing products through banner advertisements. A publisher manages a web site or an advertisement network that sells banner advertisement space. Finally, a visitor visits the web pages of a web site which contains banners.

An advertiser has the goal of spreading his/her advertisement to a certain percentage of people visiting a web site. The goal of the publisher is to make more revenue by efficiently using the advertising space available in the web pages of a web site and meeting the demands of multiple advertisers. For a given web site and period, one can analyse the visitors' behaviour by processing the transactions generated based on click stream dataset and identify the sets of web pages that cover a given percentage of visitors' population. However, the research issue here is to investigate the approaches for discovering the sets of web pages which can cover a given percentage of visitors' population based on transactions extracted from the click stream data.

Most of the research work on online advertisement has been focused on auction models [7], keyword or phrase identification based on user queries [8], contextual advertising [9] and allocation and scheduling of advertisements [10]. To our knowledge, not much amount of research work has been carried out on improving the options offered by the publisher to the advertisers.

The proposed model of coverage patterns could help the advertiser by making his advertisement visible to a certain percentage of web site visitors. With the proposed approach, it is possible to ensure that the publisher can meet the demands of multiple advertisers by considering several groups of potential pages. Through experimental results on the real world datasets we show that the proposed model and algorithm is efficient. It has a potential to improve the performance of banner advertisement placement.

A preliminary approach was presented in [11] to extract coverage patterns for banner advertisement placement. In this paper we have elaborated the model and presented a formal model of coverage patterns. We also proposed an efficient algorithm to extract complete set of coverage patterns and conducted experiments.

The rest of this paper is organized as follows: In section 2, we propose the model and approach to extract coverage patterns. In section 3, we present experimental results. In the last section, we present the conclusion and future work.

2 Model of Coverage Patterns

In this section, we first explain the model of coverage patterns. Next, we discuss the computational issues involved in extracting coverage patterns and explain how the notion of sorted closure property can be exploited for efficient extraction of coverage patterns. Subsequently, we present the algorithm to extract coverage patterns.

2.1 Coverage Patterns

As already mentioned, we identify the issue of banner advertisement placement as one of the potential application of coverage patterns. For a given e-commerce web site, the

transactions generated from click stream dataset can be used to identify the sets of web pages that cover a given percentage of visitors' population. Such a knowledge could be used to place the banner advertisements assuming similar visitors' behaviour. The related issues will be investigated as a part of future work.

To present the model of coverage patterns, we consider transactions generated from click stream data of a web site. However, the model can be extended to any transactional data set.

The basic terminology is as follows: Let $W = \{w_1, w_2, \cdots, w_n\}$ be a set of identifiers of web pages and D be a set of transactions, where each transaction T is a set of web pages such that $T \subseteq W$. Associated with each transaction is a unique transactional identifier called TID. Let T^{w_i}, $w_i \in W$ be the set of all $TIDs$ in D that contain the web page w_i. A set of web pages $X \subseteq W$ i.e., $X = \{w_p, \cdots, w_q, w_r\}$, $1 \leq p \leq q \leq r \leq n$, is called the pattern. A pattern containing k number of web pages is called a k-pattern. In other words, the length of k-pattern is k.

Example 1. *Consider the transactional database shown in Table 1. It contains 10 transactions. The set of pages, $W = \{a, b, c, d, e, f\}$. The $TIDs$ containing the web page 'a' are 1, 2, 3, 4 and 10. Therefore, $T^a = \{1, 2, 3, 4, 10\}$. The set of web pages 'a' and 'b' i.e., $\{a, b\}$ is a pattern. Since there are two web pages in this pattern it is a 2-pattern.*

Table 1. Transactional database

TID	1	2	3	4	5	6	7	8	9	10
Pages	a, b, c	a, c, e	a, c, e	a, c, d	b, d, f	b, d	b, d	b, e	b, e	a, b

The percentage of transactions in D that contain the web page $w_i \in W$ is known as the "relative frequency of a web page $w_i \in W$" and denoted as $RF(w_i)$.

Definition 1. *(Relative frequency of a web page $w_i \in W$.) Let $|T^{w_i}|$ indicates the total number of transactions that contain w_i. The relative frequency of w_i is denoted as $RF(w_i)$. That is, $RF(w_i) = \frac{|T^{w_i}|}{|D|}$.*

Note that from the advertisement point of view the pages that are visited by more number of users are interesting. We capture this aspect with the notion of frequent page. The frequent web pages are web pages which have relative frequency no less than the user-specified threshold value, called minimum relative frequency.

Definition 2. *(Frequent web page.) A web page $w_i \in W$ is considered frequent if $RF(w_i) \geq minRF$, where $minRF$ is the user-specified minimum relative frequency threshold.*

Example 2. *Continuing with the example, the relative frequency of 'a' i.e., $RF(a) = \frac{|T^a|}{|D|} = \frac{5}{10} = 0.5$. If the user-specified $minRF = 0.5$, then 'a' is called a frequent web page because $RF(a) \geq minRF$.*

Next, we capture the notion that given a set of web pages how many users visit at least one web page in the set. It means that if we place an advertisement on all pages in the set it will guarantee the delivery of advertisement to the users who visit atleast one page. This aspect is captured through the notion of *coverage set*.

Definition 3. *(Coverage set of a pattern $X = \{w_p, \cdots, w_q, w_r\}$, $1 \leq p \leq q \leq r \leq n$.) The set of distinct $TIDs$ containing at least one web page of X is called the coverage set of pattern X and is denoted as $CSet(X)$. Therefore, $CSet(X) = T^{w_p} \cup \cdots \cup T^{w_q} \cup T^{w_r}$.*

A pattern will be interesting if its coverage set contains more than a threshold number of transactions. This aspect is captured through the notion of coverage support.

Definition 4. *(Coverage-support of a pattern X.) The ratio of size of coverage set of X to the transactional database size is called the coverage-support of pattern X and is denoted as $CS(X)$.*

$$CS(X) = \frac{|CSet(X)|}{|D|}. \tag{1}$$

Example 3. *The set of web pages 'a' and 'b' i.e., $\{a, b\}$ is a pattern. The set of $tids$ containing the web page 'a' i.e., $T^a = \{1, 2, 3, 4, 10\}$. Similarly, $T^b = \{1, 5, 6, 7, 8, 9, 10\}$. The coverage set of $\{a, b\}$ i.e., $CSet(\{a, b\}) = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Therefore, coverage support of $\{a, b\}$ i.e., $CS(\{a, b\}) = \frac{|CSet(\{a,b\})|}{|D|} = \frac{10}{10} = 1$.*

For a pattern X, $CS(X) \in [0, 1]$. If $CS(X) = 0$, no single web page of X has appeared in the entire transactional database. If $CS(X) = 1$, every transaction in T contains at least one web page $w_j \in X$.

It can be noted that once a pattern X has certain coverage-support, then adding other web pages in particular web pages co-occurring with any of the web pages belonging to X to that pattern may not increase the coverage support significantly. From the advertisement point of view, such a pattern can be uninteresting to the advertiser. This is because the same users visit the web pages as there is an overlap of coverage set of X and coverage set of new single web page pattern.

Example 4. *In the transactional database shown in Table 1, $T^{\{a\}} = \{1, 2, 3, 4, 10\}$ and $T^{\{c\}} = \{1, 2, 3, 4\}$. The coverage-support of $\{a, c\}$ i.e., $CS(\{a, c\}) = \frac{5}{10} = 0.5$. If user-specified $minCS = 0.5$, then $\{a, c\}$ is an interesting pattern. However, this pattern is uninteresting as the pattern 'c' has not increased coverage-support of the pattern 'a'.*

To capture this aspect, we introduce the notion *overlap ratio of the pattern*.

Definition 5. *(Overlap ratio of a pattern.) Overlap ratio of a pattern $X = \{w_p, \cdots, w_q, w_r\}$, where $1 \leq p \leq q \leq r \leq n$ and $|T^{w_p}| \geq \cdots \geq |T^{w_q}| \geq |T^{w_r}|$, is the ratio of the number of transactions common in $X - \{w_r\}$ and $\{w_r\}$ to the number of transactions in w_r. It is denoted as $OR(X)$ and is measured as follows.*

$$OR(X) = \frac{|(T^{w_p} \cup \cdots \cup T^{w_q}) \cap (T^{w_r})|}{|T^{w_r}|} \tag{2}$$

For a pattern X, $OR(X) \in [0, 1]$. If $OR(X) = 0$, there exists no common transactions between $X - \{w_r\}$ and $\{w_r\}$. If $OR(X) = 1$, w_r has occurred in all the transactions where at least one web page $w_j \in (X - \{w_r\})$ has occurred.

Example 5. *Continuing with Example 3, the* $OR(\{a, b\}) = \frac{|CSet(b) \cap CSet(a)|}{|CSet(a)|} = \frac{2}{5} = 0.4$.

Note that a coverage pattern is interesting if it has high coverage support and low overlap ratio. As a result an advertisement is exposed to more number of users by reducing repetitive display of the advertisement. The definition of coverage pattern is as follows.

Definition 6. *(Coverage pattern X.) A pattern X is said to be a coverage pattern if $CS(X) \geq minCS$, $OR(X) \leq maxOR$ and $RF(w_i) \geq minRF$, $\forall w_i \in X$. The variables, $minCS$ and $maxOR$ represent user-specified minimum coverage support and maximum overlap ratio, respectively. A coverage pattern X having $CS(X) = a\%$ and $OR(X) = b\%$ is expressed as*

$$X \quad [CS = a\%,\ OR = b\%] \tag{3}$$

Example 6. *If $minRF = 0.4$, $minCS = 0.7$ and $maxOR = 0.5$, then the pattern $\{a, b\}$ is a coverage pattern. It is because $RF(a) \geq minRF$, $RF(b) \geq minRF$, $CS(\{a, b\}) \geq minCS$ and $OR(\{a, b\}) \leq maxOR$. This pattern is written as follows:*

$$\{a, b\} \quad [CS = 1\ (= 100\%),\ OR = 0.4\ (= 40\%)]$$

Problem statement: Given a transactional database D, set of web pages W, and user-specified minimum relative frequency ($minRF$), minimum coverage support ($minCS$) and maximum overlap ratio ($maxOR$), discover complete set of coverage patterns such that

i. If X is a coverage 1-pattern (i.e., $k = 1$), then $RF(w_i) \geq minRF$ and $RF(w_i) \geq minCS$, $\forall w_i \in X$.
ii. Otherwise (i.e., when $k > 1$), each coverage pattern X must have $CS(X) \geq minCS$, $OR(X) \leq maxOR$ and $RF(w_i) \geq minRF$, $\forall w_i \in X$.

2.2 Mining Coverage Patterns

A naive approach to find the complete set of coverage patterns for a dataset consisting of n web pages is to generate all possible $(2^n - 1)$ combinatorial patterns (CP) from n web pages. Now, each pattern in CP is added to the coverage pattern set if it satisfies $minCS, minRF$ and $maxOR$ constraints. The problem with this approach is, if n is large the search space will be large leading to high computational cost. The search space can be reduced if the coverage pattern satisfies downward closure property on either coverage support or overlap ratio.

Our analysis on coverage patterns states that the measure *coverage support* does not satisfy *downward closure property*. That is, although a pattern satisfies $minCS$, it is not necessary that all its non-empty subsets will also satisfy $minCS$ value.

Example 7. *Consider the patterns* $\{a\}$, $\{e\}$ *and* $\{a, e\}$. *The coverage supports of these patterns are 0.5, 0.4 and 0.7, respectively. If the user-specified* $minCS = 0.7$, *then the pattern* $\{a, e\}$ *satisfies* $minCS$ *value. However, its non-empty subsets do not satisfy* $minCS$ *value.*

The parameter *overlap ratio* also does not satisfy *downward closure property* if a pattern is considered as an unordered set of web pages. However, this measure satisfies *downward closure property* if a pattern is an ordered set, where web pages are sorted in descending order of their frequencies. This property is known as the *sorted closure property* [12].

Property 1. If $X \subset Y$, then $CSet(X) \subseteq CSet(Y)$.

Property 2. **Sorted closure property**: Let $X = \{w_p, \cdots, w_q, w_r\}$ be a pattern such that $RF(w_p) \geq \cdots \geq RF(w_q) \geq RF(w_r)$ and $1 \leq p \leq q \leq r \leq n$. If $OR(X) \leq maxOR$, all its non-empty subsets containing w_r and having size $k \geq 2$ will also have overlap ratio less than or equal to $maxOR$.

Rationale: Let w_a, w_b and w_c be the web pages having $RF(w_a) \geq RF(w_b) \geq RF(w_c)$. If $OR(w_a \cup w_c) > maxOR$, then $OR(\{w_a \cup w_b\} \cup w_c) > maxOR$ because from $Property$ 1

$$\frac{|CSet(w_a) \cap CSet(w_c)|}{|CSet(w_c)|} \leq \frac{|CSet(\{w_a \cup w_b\}) \cap CSet(w_c)|}{|CSet(w_c)|} \tag{4}$$

Definition 7. *(Non-overlap pattern* X*.) A pattern* X *is said to be* non-overlap *if* $OR(X) \leq maxOR$ *and* $RF(w_i) \geq minRF$, $\forall w_i \in X$.

Every coverage pattern is a non-overlap pattern, however it is not the same vice versa. The *sorted closure property* of non-overlap patterns is used for minimizing the search space while mining complete set of coverage patterns by designing an algorithm similar to the Apriori algorithm [1]. The detailed algorithm for mining the complete coverage patterns is given in next subsection.

2.3 Coverage Pattern Extraction Algorithm

We use the following notations. Let F be a set of frequent items, C_k be a set of candidate k-patterns, L_k be a set of coverage k-patterns and NO_k be a set of non-overlap k-patterns. The proposed algorithm *CMine* employs a *level-wise* search to discover the complete set of coverage patterns. In *level-wise* search, k-patterns are used to explore $(k+1)$-patterns. The proposed CMine algorithm is different from Apriori algorithm [1] used for mining frequent patterns. The main reason is as follows: Frequent patterns satisfy *downward closure property*. Therefore, Apriori algorithm uses frequent k-patterns to explore $(k + 1)$-patterns. CMine cannot explore $(k + 1)$-patterns with coverage k-patterns as coverage patterns no longer satisfy *downward closure property*.

The detailed description of the algorithm is as follows: The algorithm CMine begins with a scan of the database and discovers set of all frequent web pages (denoted as F) and coverage 1-patterns (denoted as L_1). Non-overlap 1-patterns (denoted as NO_1) will be the set of all frequent 1 web pages. Next, web pages in NO_1 are sorted in descending order of their frequencies. This is an exception from Apriori algorithm [1] that has to be carried out in CMine algorithm to efficiently mine coverage patterns. Each web page $w_i \epsilon\ NO_1$ is of the form $< w_i, T^{w_i} >$ where T^{w_i} denote set of transaction ids which contain the web page w_i. Using NO_1 as a *seed set*, candidate patterns C_2 are generated by combining $NO_1 \bowtie NO_1$. From C_2, the patterns that satisfy $minCS$ and $maxOR$ are generated as coverage 2-patterns, L_2. Simultaneously, all candidate 2-patterns that satisfy $maxOR$ are generated as non-overlap 2-patterns NO_2. Since overlap patterns satisfy *sorted closure property*, C_3 is generated by combining $NO_2 \bowtie NO_2$. From C_3, L_3 and NO_3 are discovered. At each level 'k', a two-step process is followed, consisting of join and prune actions [13].

1. The join step: To find L_k, a set of candidate k-web page sets C_k is generated by joining NO_{k-1} with itself. Let l_1 and l_2 be web page sets in NO_{k-1}. Note that the members of NO_{k-1} are join-able if their first $(k-2)$ web pages are in common.
2. The prune step: C_k is a superset of L_k, that is, its members may or may not be coverage patterns, but all of the coverage k-web page sets are included in C_k. The number of k-web page sets in C_k, however, can be huge, and so this could involve heavy computation. To reduce the size of C_k, the Sorted closure property of non-overlap patterns is used as follows. Any $(k-1)$-web page set that is not satisfying the overlap ratio cannot be a subset of a non-overlap k-web page set. Hence, if any $(k-1)$-ordered subset of a candidate k-web page set is not in NO_{k-1}, then the candidate cannot be a non-overlap pattern either and so can be removed from C_k. This pruning step is used to reduce the search space.

The above process is repeated until no new coverage pattern is found or no new candidate pattern can be generated.

The proposed algorithm uses bitwise operations to find the complete set of coverage patterns. So, a single scan of the database (to find the bit strings for all single web page sets) is sufficient for the algorithm to find the complete set of coverage patterns. Generation of bit strings for larger web page sets and computation of CS, OR for a web page set can be carried out by using simple bitwise AND and OR operations which makes the algorithm computationally very fast.

We now explain the working of *CMine* algorithm using the transactional database, T, shown in Table 1. There are 10 transactions in this database, that is, $|T| = 10$. Let the user-specified $minRF$, $minCS$ and $maxOR$ be 0.4, 0.7 and 0.5, respectively. The column titled *Bitstring* represents binary representation of coverage set of pattern i. For example, the bit string corresponding to the pattern "{b,a}" is "1111111111". This implies that every transaction of T contains either b or a or both. For binary representation of TID's, union of coverage sets of two patterns is equal to boolean OR operation of corresponding bit strings. Similarly, intersection of coverage sets of two patterns is equal to boolean AND operation of corresponding bit strings. We use Figure 1 to illustrate the CMine algorithm for finding coverage patterns in T.

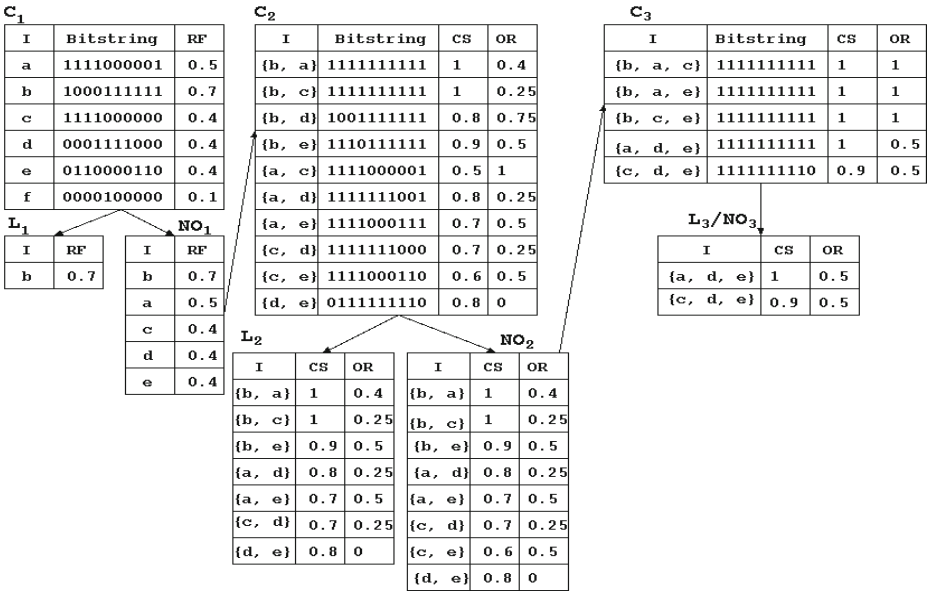


Fig. 1. Working of CMine algorithm. The term 'I' is an acronym for item set.

i. The algorithm *CMine* scans all the transactions to generate bit string B^{w_i} and relative frequencies (RF) of each web page $w_i \epsilon T$. $RF(w_i) = \frac{|B^{w_i}|}{|T|}$. $|B^{w_i}|$ denotes the number of 1's in the bit string. Each web page, $w_i \epsilon T$ is a member of the set of candidate 1-pattern, C_1.

ii. From C_1, the set of coverage 1-patterns, L_1, are discovered if their frequencies are greater than or equal to $minCS$. Simultaneously, set of non overlap 1-patterns, NO_1, are discovered if candidate 1-patterns have relative support greater than or equal to $minRF$ and finally the web pages in NO_1 are sorted in the decreasing order of their frequencies.

iii. To discover the set of coverage 2-patterns, L_2, the algorithm computes the join of $NO_1 \bowtie NO_1$ to generate a candidate set of 2-patterns, C_2.

iv. Using Equation 1, *coverage support* of each candidate pattern is computed by boolean OR operation. For example, CS(b,a) = $\frac{|B^b \vee B^a|}{|T|} = \frac{|1111111111|}{10} = \frac{10}{10} = 1.0$. Next, *overlap ratio* for each candidate pattern is computed by boolean AND operation. For example, OR(b,a) = $\frac{|B^b \wedge B^a|}{|B^a|} = \frac{|1000000001|}{5} = \frac{2}{5} = 0.4$. The columns titled '$CS$' and '$OR$' respectively show the *coverage support* and *overlap ratio* for the patterns in C_2.

v. The set of candidate 2-patterns that satisfy $maxOR$ are discovered as non-overlap 2-patterns, denoted as NO_2. Simultaneously, the set of candidate 2-patterns that satisfy both $minCS$ and $maxOR$ are discovered as coverage 2-patterns.

vi. Next, C_3 is generated by $NO_2 \bowtie NO_2$. That is, $C_3 = NO_2 \bowtie NO_2 = \{\{b, a, c\}, \{b, a, e\}, \{b, c, e\}, \{a, d, e\}, \{c, d, e\}\}$.

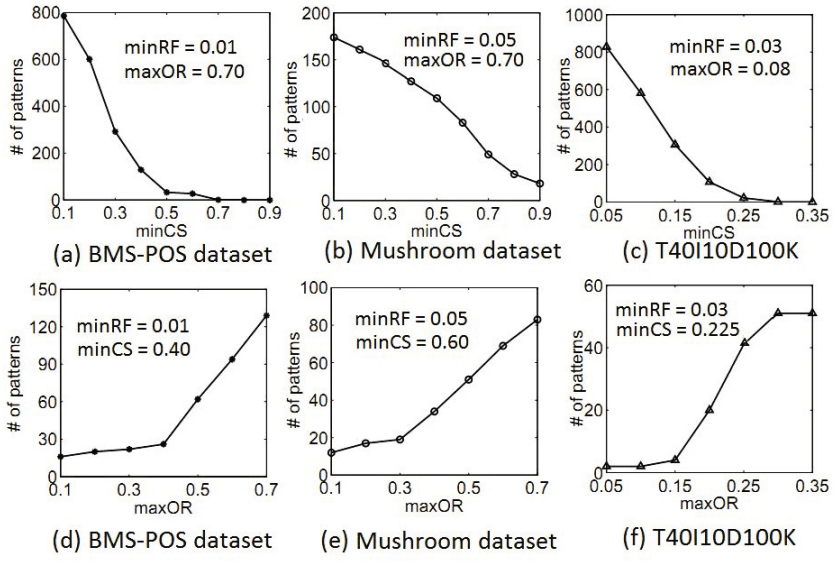


Fig. 2. Number of patterns generated by CMine algorithm at different $minCS$ and $maxOR$ values for BMS-POS, Mushroom and T40I10D100K datasets

vii. As in step v, we discover non-overlap 3-patterns, NO_3, and coverage 3-patterns, L_3. The algorithm stops as no more candidate 4-patterns can be generated from non-overlap 3-patterns.

3 Experimental Results

For experimental purposes we have chosen four real world datasets and one synthetic dataset. The detailed description of the datasets are given below.

i. Kosarak dataset is a sparse dataset with 990,002 number of transactions containing 41,270 distinct items [14].
ii. MSNBC dataset contains data from Internet Information Server (IIS) logs for msnbc.com and news-related portions of msn.com for the entire day of September, 28, 1999 [15]. Requests are at the level of page category. The number of categories are 17 and the number of transactions are 989,818.
iii. Mushroom dataset is a dense dataset containing 8,124 transactions and 119 distinct items [14].
iv. BMS-POS dataset contains click stream data of a dotcom company [14]. The dataset contains 515,597 number of transactions and 1656 distinct items.
v. The synthetic dataset T40I10D100K is generated by the dataset generator [16]. The dataset contains 100,000 transactions and 941 distinct items.

The CMine algorithm was written in Java and run with Windows XP on a 2.66 GHz machine with 2GB memory.

3.1 Coverage Pattern Generation

The Figure 2(a) shows the number of patterns generated (y-axis) for BMS-POS dataset for different values of *minCS* (x-axis) while *minRF* and *maxOR* are fixed at the values 0.01 and 0.7 respectively. It can be observed from the Figure 2(a) that the number of coverage patterns decrease with the increase in *minCS*, and more importantly, the number of patterns generated are very few when *minCS* is greater than 0.5. In general, for a given *maxOR*, coverage support of coverage patterns increases with the length of the pattern due to the addition of new frequent items. The length of a coverage pattern increases with increasing levels of iteration for generation of candidate itemsets in CMine algorithm. However for higher levels of iteration due to overlap ratio constraint, the number of non-overlap patterns generated is decreased. Therefore, the number of coverage patterns generated having higher coverage support decreases with increasing *minCS*. The relation between the number of coverage patterns generated and *minCS* which was apparent in Figure 2(a) is also observed for Figure 2(b) and 2(c). It can also be observed form Figure 2(c) that no coverage patterns are generated for minCS = 0.35. This implies, that maximum threshold of *minCS* for *minRF* = 0.03, *maxOR* = 0.8 for T40I10D100k dataset is 0.35 since no coverage patterns are generated for minCS greater than 0.35.

The Figure 2(d) shows number of patterns generated (y-axis) for BMS-POS dataset for different values of *maxOR* (x-axis) while *minRF* and *minCS* are fixed at the values 0.01 and 0.40 respectively. It can be observed from Figure 2(d) that the gradient of the curve increases linearly from *maxOR* = 0.1 to 0.4. For maxOR = 0.4, the gradient of the curve changes and again increases linearly from *maxOR* = 0.4 to 0.7. However, the gradient of the curve from *maxOR* = 0.4 to 0.7 is greater than the curve for *maxOR* = 0.1 to 0.4. This implies that the number of patterns generated increases with increasing *maxOR* value. As the *maxOR* value is increased, the number of items of the candidate sets C_i (i =2,3,4...k-1) are increased which will result in increase of number of coverage patterns generated. Similar to the Figure 2(d), the phenomenon of increase in generation of coverage patterns with respect to *maxOR* value is also observed in Figure 2(e) and 2(f). It can be observed form Figure 2(f) that the number of patterns generated become constant for *maxOR* $\geq$ 0.30. This implies that no new nonoverlap patterns are generated for higher levels of iteration for generation of candidate item sets in CMine algorithm such that coverage patterns extracted from non-overlap patterns have coverage support greater than 0.225.

3.2 Scalability Experiment

We used *Kosarak* dataset to conduct scalability experiment. We divided the dataset into five portions of 0.2 million transactions in each part. We investigated the performance of CMine Algorithm after cumulatively adding each portion with previous parts and extracting coverage patterns each time. The values of *minRF*, *minCS* and *maxOR* are fixed at 0.01, 0.1 and 0.5 respectively. The experimental results are shown in Figure 3. It is clear from the Figure 3 that as the database size increases, the execution time also increased linearly.

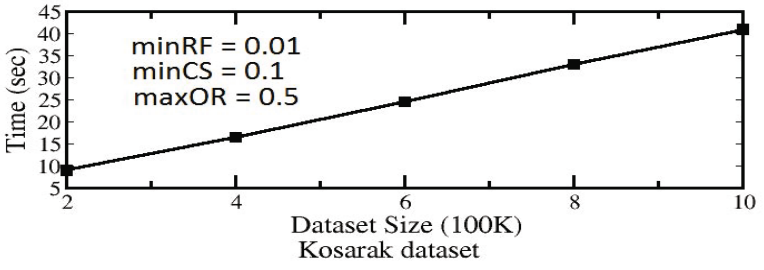


Fig. 3. Scalability of CMine algorithm

3.3 Usefulness of Coverage Patterns

Table 2 shows some coverage patterns generated by *Cmine* algorithm for $minCS = 0.4$ and $maxOR = 0.5$ and $minRF = 0.02$ for MSNBC dataset. The names of web page categories involved in MSNBC are "frontpage", "news", "tech", "local", "opinion", "on-air", "misc", "weather", "health", "living", "business", "sports", "summary", "bbs" (bulletin board service), "travel", "msn-news", and "msn-sports". From Table 2, it can be observed that any of the six coverage patterns ensure about 40 percent coverage. The result indicates how the proposed approach provides flexibility to the publisher to meet the demands of multiple advertisers by considering different sets of web pages.

Table 2. Sample coverage 3-patterns extracted from MSNBC dataset [15]

S.No	Coverage Pattern	CS	S.No	Coverage Pattern	CS
1	$\{local, misc, frontpage\}$	0.42	4	$\{on\text{-}air, news, misc\}$	0.40
2	$\{news, health, frontpage\}$	0.43	5	$\{tech, weather, on\text{-}air\}$	0.41
3	$\{tech, opinion, frontpage\}$	0.41	6	$\{sports, misc, opinion\}$	0.43

4 Conclusions and Future Work

In this paper we have proposed a new data mining pattern called "coverage pattern" and proposed an efficient methodology to extract the same from transactional databases. We have explained how coverage patterns could be useful by considering the issue of banner advertisement placement. By conducting experiments on different kinds of datasets, we have shown that the proposed model and methodology can effectively discover coverage patterns.

As a part of the future work, we are going to investigate how both frequent and coverage pattern knowledge can be used for efficient banner advertisement placement. In addition we are planning to investigate how the content of the web page and search query can be exploited to explore content specific coverage patterns. We are also exploring how the notion of coverage patterns can be extended to other domains like bio-informatics for extracting potential knowledge patterns.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB 1994: Proceedings of the 20th International Conference on Very Large Data Bases, pp. 487–499. Morgan Kaufmann Publishers Inc. (1994)
2. Chvatal, V.: A greedy heuristic for the set-covering problem. Mathematics of Operations Research, 233–235 (1979)
3. Garey, M.R., Johnson, D.S., Stockmeyer, L.: Some simplified np-complete problems. In: Proceedings of the Sixth Annual ACM Symposium on Theory of Computing, STOC 1974, pp. 47–63. ACM (1974)
4. Venetis, P., Koutrika, G., Garcia-Molina, H.: On the selection of tags for tag clouds. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM 2011, pp. 835–844. ACM (2011)
5. Bonchi, F., Castillo, C., Donato, D., Gionis, A.: Topical query decomposition. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008, pp. 52–60. ACM (2008)
6. Amiri, A., Menon, S.: Efficient scheduling of internet banner advertisements. ACM Trans. Internet Technol. 3(4), 334–346 (2003)
7. Ghosh, A., Rubinstein, B.I., Vassilvitskii, S., Zinkevich, M.: Adaptive bidding for display advertising. In: WWW 2009: Proceedings of the 18th International Conference on World Wide Web, pp. 251–260. ACM (2009)
8. Wu, X., Bolivar, A.: Keyword extraction for contextual advertisement. In: WWW 2008: Proceeding of the 17th International Conference on World Wide Web, pp. 1195–1196. ACM (2008)
9. Chakrabarti, D., Agarwal, D., Josifovski, V.: Contextual advertising by combining relevance with click feedback. In: WWW 2008: Proceeding of the 17th International Conference on World Wide Web, pp. 417–426. ACM (2008)
10. Alaei, S., Arcaute, E., Khuller, S., Ma, W., Malekian, A., Tomlin, J.: Online allocation of display advertisements subject to advanced sales contracts. In: ADKDD 2009: Proceedings of the Third International Workshop on Data Mining and Audience Intelligence for Advertising, pp. 69–77. ACM (2009)
11. Sripada, B., Reddy, P.K., Kiran, R.U.: Coverage patterns for efficient banner advertisement placement. In: WWW (Companion Volume), pp. 131–132 (2011)
12. Liu, B., Hsu, W., Ma, Y.: Mining association rules with multiple minimum supports. In: KDD 1999: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 337–341. ACM (1999)
13. Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann (2006)
14. Fimi: Frequent itemset mining implementations repository (July 2010), http://fimi.cs.helsinki.fi/
15. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml
16. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, SIGMOD 1993, pp. 207–216. ACM (1993)

Discovering Unknown But Interesting Items on Personal Social Network

Juang-Lin Duan, Shashi Prasad, and Jen-Wei Huang

Yuan Ze University,
135 Yuan-Tung Road, Chung-Li, Taiwan 32003, R.O.C.
{s986039,s989112}@mail.yzu.edu.tw,
jwhuang@saturn.yzu.edu.tw

Abstract. Social networking service has become very popular recently. Many recommendation systems have been proposed to integrate with social networking websites. Traditional recommendation systems focus on providing popular items or items posted by close friends. This strategy causes some problems. Popular items always occupy the recommendation list and they are usually already known by the user. In addition, items recommended by familiar users, who frequently communicate with the target user, may not be interesting. Moreover, interesting items from similar users with lower popularity are ignored. In this paper, we propose an algorithm, UBI, to discover unknown but interesting items. We propose three scores, i.e., Quartile-aided Popularity Score, Social Behavior Score, and User Similarity Score, to model the popularity of items, the familiarity of friends, and the similarity of users respectively in the target user's personal social network. Combining these three scores, the recommendation list containing unknown but interesting items can be generated. Experimental results show that UBI outperforms traditional methods in terms of the percentages of unknown and interesting items in the recommendation list.

Keywords: recommendation, social network, unknown but interesting.

1 Introduction

With the tremendous success of social networking websites nowadays, diverse social network services have been vigorous and much popular. Although many services related to social network websites exist, it is important to set up a standard which helps users to make a decision if it is worthy of them. In recent years, many recommendation systems have been proposed to integrate with social networking websites and been used in many different business applications such as movies, music, books, news, etc. Some popular e-commerce websites such as Amazon, eBay, and Netflix analyze the shopping behavior of users to build the recommendation list of products to their customers. The online shopping system recommends each customer the products bought by others who have similar shopping behavior in the past. In addition, some well-known social networking websites, Facebook, Myspace and Twitter provide users to establish

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 145–156, 2012.

their own personal communities or social networks based on friends. There are many services for personalized recommendations in social networking websites. For example, InSuggest[1] provides personalized recommendations of bookmarks originating from the social bookmarking site Delicious[2], and Outbrain[3] provides personalized blog recommendations from blogging services. The purpose of these recommendations are to adapt the contents of the websites to the specific needs of the individual user by presenting the most attractive and relevant items to users.

Traditional recommendation systems usually generate recommendation lists based on popularity of items and/or analyze the behavior of the target user and then make further recommendations. However, these systems focus on providing popular items or items posted by close friends. This leads to problems listed as follows:

1. Popular items always occupy the recommendation list and they are usually already known by the user.
2. Items recommended by familiar users, who frequently communicate with the target user, may not be interesting.
3. Interesting items from similar users with lower popularity are ignored.

Fig. 1 shows the personal social network of the target user U_1. Circular nodes represent users and the link between two users indicates the friend relationship. The number on the link denotes the number of direct communication between these two users and MF represents the number of mutual friends between U_1 and the other user. In addition, square nodes represent items posted by the connected user, where the number indicates the number of comments left by all users and the number of likes, which means other users are interested in the item. Traditional recommendation systems usually recommend items $I_4 : 150$ and $I_1 : 100$, since I_4 is the most popular item and I_1 is copied by many users. However, these items are easily found by the target user and should not occupy the recommendation list. On the other hand, $I_9 : 55$ from similar user U_3 with lower popularity is easily ignored. New paragraph to remedy these defects, we propose to discover unknown but interesting items, and design an algorithm to generate recommendation list on personal social networks. The personal social network contains the target user and his/her direct friends. We also include items posted by these users.

Our proposed algorithm not only considers the popularity of items and the similarity with friends, but also discovers unknown but interesting items through the target user's social behavior. We propose three scores to calculate Unknown But Interesting Score of each item in the target user's personal social network. The first score is Quartile-aided Popularity Score, which is based on the popularity adjusted by quartiles of items, to find out items with lower popularity. The

[1] http://insuggest.wordpress.com/
[2] http://www.delicious.com/
[3] http://www.outbrain.com/

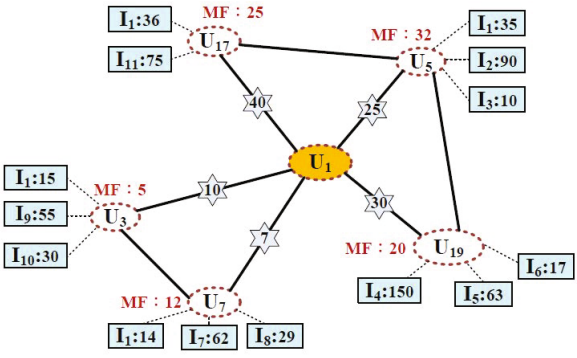


Fig. 1. The personal social network of the target user U_1

second score is Social Behavior Score, which depends on social interactions on social network websites and direct communication between users. The third score is User Similarity Score, which is based on interests between users, to model the similarity of the target user and his/her friends. Combining these three scores, we can generate the recommendation list with unknown but interesting items to the target user.

Finally, to evaluate our approach, we implement our system on a social networking website, and collect users' feedback to compare differences between traditional approaches and our proposed algorithm. The experimental results show that our proposed algorithm can successfully find unknown but interesting items and the satisfaction percentage of our system is higher than compared methods.

The remainder of this paper is organized as follows. In Section 2, a brief survey of related works is presented. The proposed algorithm is introduced in Section 3. Section 4 presents the experimental results. Finally, we conclude this work in Section 5.

2 Related Works

2.1 Social Networking

Social networking is the development of social collaborative technologies, and connected by one or more specific types of relationship, such as friendship, similar interest. In recent year, online social networking has been around in the world, therefore, many online social networking websites are being generated which allow users to establish their social network by adding other users to their friend lists. For example, many users of popular social networking sites such as Facebook and Twitter. Many research issues of social networking focused on development of information techniques and data processing [7], and then extend to social networking analysis [10], which is a set of methods to discover relations between nodes in a social network.

A number of measures in social networking analysis including network size, degree centrality, betweenness, density etc. are considered. Bird *et al.* [5] proposed a method to extract social networks from e-mail communication. Agrawal *et al.* [3] using web mining techniques to understand the behavior of users in news group, the proposed the behavior is meaning a newsgroup posting consists of one or more quoted lines from another posting followed by the opinion of the author. Adamic *et al.* [2] developed a method to discover the relationship of friends and neighbors in the web. Many social networking analysis approaches have propose similar ideas to find neighborhoods and paths with the social network [8], [9]. In our work, we extend the concept of social networking to discover the unknown but interesting items from social network site.

2.2 Recommendation Systems

Recommendation systems are widely used for personalized information filtering technology, always used to recommend items that are of interest to users based on customer demographics, features of items, or user preferences. Therefore, users should provide their interest profiles to recommendation systems in order to get recommendations. Then recommendation systems can utilize these interest profiles to estimate the ratings of the unrated items for users or predict that items to be liked by users. In general, recommendation systems are usually classified into the following three methods: content-based recommendation, collaborative filtering and hybrid approaches.

The first method is based on contents [12], which analyzes the contents of information products and user information to produce the recommended method. This method is mainly dependent on the data description of goods and users of consumer behavior in the past, for the two meta-analysis to calculate the characteristics of different commodities of the scores for the summary, identify the items for the user with a higher satisfaction scores in order to establish recommended.

The second method is based on collaborative filtering [14], which utilizes similarities of user's preferences to recommend items. Collaborative filtering is a set of similarity measure methods, as follows: Jaccard's coefficient of similarity, Cosine similarity [15], Pearson correlation-based similarities [13]. Many approaches employ the technique of collaborative filtering, for instance, Bell *et al.* [4] proposed novel algorithms for predicting user ratings of items by integrating complementary models that focus on patterns at different scales. Facebook has a feature, called "People You May Know", which recommends user to connect with based on a "friend of a friend" approach [1].

Finally, many recommendation systems use hybrid approaches by combining content-based methods and collaborative filtering [6], which helps to avoid certain limitations of content-based and collaborative systems. For example, TANGENT [11] focused on the "surprise me" query, in which a user may be bored with usual genre of items, and may recommend new genre of items. This research

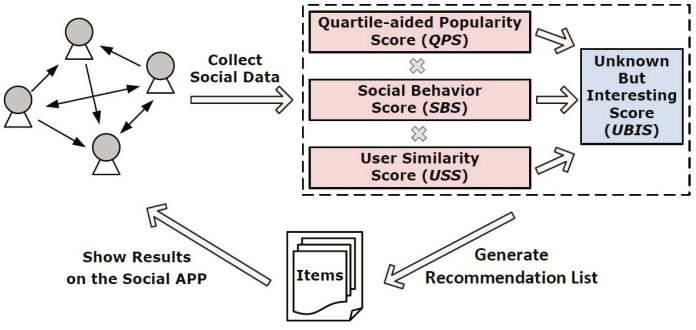


Fig. 2. The architecture of the system

closes to our belief, however, traditional recommendation systems always focus on high frequency of item with similarity of user. That gives us an inspiration that we can make use of the impact in our work.

3 Unknown But Interesting Recommendation System

In this section, we illustrate the system architecture and the details of our proposed algorithm, Unknown But Interesting algorithm.

3.1 System Architecture

We build a recommendation system on a social networking platform. The architecture of the system is shown in Fig. 2. First, when the target user logs in to our system, the target user's social data is collected and is used to generate the personal social network. The social data includes the target user's profile, which contains his/her posts, interests, friends list, and social interactions. From the friends list, the open information of the target user's friends is also collected, which includes the number of mutual friends and direct communication. We only consider 1-level friends, who have the direct connections to the target user. The reason is that users are usually not interested in the social behavior of friends of 1-level friends and friends of friends have little influence power to the target user. After the collection of the social data, we analyze the personal social network to calculate three different scores of each item and obtain unknown but interesting score by the proposed algorithm. Finally, we can generate the recommendation list of items show the results on the social networking website.

3.2 Unknown But Interesting Algorithm

Unknown But Interesting (UBI) algorithm focuses on the popularity of items, the similarity between users, and the social behavior of the target user. The proposed algorithm calculate three scores as follows:

iv	I_3U_5	I_1U_7	I_1U_3	I_6U_{19}	I_8U_7	$I_{10}U_3$	I_1U_5	I_1U_{17}	I_9U_3	I_7U_7	I_5U_{19}	$I_{11}U_{17}$	I_2U_5	I_4U_{19}
PS_{iv}	10	14	15	17	29	30	35	36	55	62	63	75	90	150
QS_{iv}	53	49	48	46	34	33	28	27	8	1	0	12	27	87
QPS_{iv}	0.40	0.44	0.45	0.48	0.61	0.63	0.68	0.69	0.91	0.99	1	0.86	0.69	0.01

Fig. 3. Example of PS_{iv}, QS_{iv}, and QPS_{iv}

1. Quartile-aided Popularity Score (QPS) of each item.
2. Social Behavior Score (SBS) by considering social interactions of users.
3. User Similarity Score (USS) of each friend of the target user.

Finally, we combine these three scores to obtain the Unknown But Interesting Score ($UBIS$), and provide the recommendation list to the target user. We explain the formulas and significance of each score as follows.

Quartile-Aided Popularity Score. In the social networking website, users can post messages, photos, videos, and links on their own pages. Other users are able to leave comments on the posted items or simply click "like" button to show their interests in the items. Therefore, in our system, the popularity score, PS_{iv}, is defined as the number of comments and likes of a certain item i posted by user v.

As we explained earlier, traditional recommendation systems always recommend popular items to users, but these items are well-known and easily noticed by users themselves. In order to determine a certain degree of popular items, we use the concept of quartile, $Q_r = |r\,(n+1)\,/4|$, where r determines which quartile, and n is total number of items. The items are sorted by their popularity score, PS_{iv}, ascendingly. The upper quartile, $Q_3(PS_{iv})$, represents the popularity of the item with the $\lfloor 3\,(n+1)/\,4 \rfloor$th rank in the list. In this way, we define quartile score, QS_{iv}, to be the popularity score minus the upper quartile.

$$QS_{iv} = PS_{iv} - Q_3\,(PS_{iv}) \tag{1}$$

Fig. 3 shows the PS_{iv} and QS_{iv} of each item posted by each user in Fig. 1. In this example, n is 14. Therefore Q_3 is the 11th lowest value of PS_{iv}. The respective QS_{iv} is shown in middle row. Furthermore, in order to find items which are not very popular but still have enough attention, we propose Quartile-aided Popularity Score, QPS_{iv}.

$$QPS_{iv} = 1 - \frac{QS_{iv}}{Max\,(QS_{iv}) + 1} \tag{2}$$

QPS_{iv} normalizes QS_{iv} by the maximum value and gives the upper quartile the highest credit. In this way, we can capture the popularity of items adjusted by the upper quartile.

Social Behavior Score. In addition to adjusting the popularity of items, UBI considers the social behavior of the target user to further discover unknown

uv	U_1U_3	U_1U_7	U_1U_{19}	U_1U_5	U_1U_{17}
$MF_{(uv)}$	5	12	20	32	25
$DC_{(uv)}$	10	7	30	25	40
$SBS_{(uv)}$	0.64	0.53	0.1	0.01	0.007

Fig. 4. Example of MF_{uv}, DC_{uv} , and SBS_{uv}

items. In the social networking website, users usually allowed to meet friends and make connections to one another. Users can easily get information from the friends they are familiar with. Therefore, UBI includes two factors from the social behavior, i.e., mutual friends and direct communication. We define mutual friend, MF_{uv}, be the number of mutual friends between user u and user v, and direct communication of users, DC_{uv}, be the number of direct communication between user u and user v. For the target user u, the more mutual friends u and v have the more likely it is that items are spread between those friends. In addition, the more direct communication there is between u and v, the more likely it is that items posted by user v are already known by the target user u. Therefore, in order to find unknown items, we define Social Behavior Score as follows.

$$SBS_{uv} = \left(1 - \frac{MF_{uv}}{\underset{v \in friends\ of\ u}{Max}(MF_{uv}) + 1}\right) \times \left(1 - \frac{DC_{uv}}{\underset{v \in friends\ of\ u}{Max}(DC_{uv}) + 1}\right), \tag{3}$$

where $Max(MF_{uv})$ represents the maximum value of MF_{uv} among all friends v of the target user u, and $Max(DC_{uv})$ is the maximum value of DC_{uv} among all v. SBS_{uv} represents the inverse probability that the items posted by user v are already known by the target user u. Fig. 4 shows some SBS_{uv} of users in Fig. 1.

User Similarity Score. UBI not only takes popularity of items and familiarity of users into consideration, but also includes the similarity of users to obtain interesting items. If the item is recommended by a similar user of the target user, it is more likely that the target user is interested in the item. At first, users can do different actions in the social networking website to show their interest in some items, e.g., posting a link, commenting on a photo, or liking a video. We give each action a worth value, WV, indicating how much a user is interested to some item by performing this action.

$$WV_j = \frac{\sum_{j \in all\ actions} times\ of\ action\ j}{times\ of\ action\ j} \tag{4}$$

For example, the number of articles is 100, the number of comments is 500, and the number of likes is 1000. We can get the sum of all action as 1600, and we can

	I_1	I_2	I_3	I_4	I_5
U_1	post+like	like+comment	like	comment	n/a
U_3	post	post+comment	comment	post+like +comment	post
U_{19}	n/a	like+comment	n/a	like	like

	I_1	I_2	I_3	I_4	I_5
U_1	17.6	4.8	1.6	3.2	0
U_3	16	19.2	3.2	20.8	16
U_{19}	0	4.8	0	1.6	1.6

Fig. 5. The score of user's behavior

calculate the worth value of posting an article to be 16, leaving a comment to be 3.2, and liking an item to be 1.6. Then, users may have a variety of behavior on the same item, as shown in Fig. 5. Therefore, we sum up the worth value of all actions performed on the same item i to get the interesting score, IS_i.

$$IS_i = \sum_{j \in all\ actions} WV_j \tag{5}$$

Finally, we can define the user behavior as

$$UB_u = \{IS_{I_1}, ..., IS_{I_n}\}, \tag{6}$$

where IS_i is the total interesting score of the user v to the item i. Accordingly, the User Similarity Score, USS_{uv}, between user u and user v is computed by the following equation.

$$USS_{uv} = \frac{UB_u \cdot UB_v}{\|UB_u\| \, \|UB_v\|} \tag{7}$$

From the user behavior listed in Fig. 5, user similarity score between U_1 and U_3 is 0.5, and USS between U_1 and U_{19} is 0.21.

Unknown But Interesting Score. Finally, we combine QPS, SBS, and USS to calculate the unknown but interesting score for each item on the personal social network of the target user u. Thus, we define Unknown But Interesting Score as follows.

$$UBIS_i = \sum_v (QPS_{iv} \times SBS_{uv} \times USS_{uv}) \tag{8}$$

where $\sum$ is the sum of same item i among all user v. Consequently, we can generate the recommended list based on $UBIS$. As shown in Fig. 6, we sort $UBIS$ and recommend the Top-k items to the target user.

i	I_9	I_{10}	I_7	I_8	I_5	I_1	I_6	I_2	I_3	I_{11}	I_4
$UBIS_i$	0.29	0.20	0.15	0.09	0.07	0.04	0.03	0.006	0.004	0.0006	0

Fig. 6. Example of $UBIS_i$

4 Experiments

In this section, the methodology and the performance evaluation are discussed. The experiment is conducted to measure the percentages of unknown but interesting items in the recommendation list. The methodology is discussed in Section 4.1. The performance evaluation is presented and discussed in Section 4.2.

4.1 Methodology

We implement recommendation system on a popular social networking website, Facebook, in order to compare our algorithm to traditional recommendation systems. We can obtain user's social information easily to discover unknown items. We generate three recommendation lists each on Facebook, traditional method, and our algorithm. First, Facebook recommendation list is based on latest updates from user's posting. Second, traditional method is based on popular items with user's preferences on Facebook. In other words, users usually focus on popularity of items with similarity among users. At last, our algorithm is based on $UBIS$ which recommends unknown but interesting items. We generate recommendation list which presents Top-20 items to the target user, as shown in the Fig.7, which is the interface of UBI recommendation system on Facebook, and we show one of the lists randomly. Furthermore, we show posted user's name, content of the item, and each list has two questions for each message, and questionnaires, which are as follows: unknown or known, and interesting. The question about unknown or known represents whether the message is unread or read by the target user respectively. The question about interesting denotes whether the target user is interested in the message. We can compare UBI algorithm with the other two methods based on our questionnaires.

4.2 Performance Evaluation

We randomly invited 355 users to participate in our experiment. Our experiments were conducted in the months starting from July through September of 2011, and 185 active users participated per month. At first, we compared three recommendation lists, and focused on unknown and interesting questions checked by users. In other words, these two indicators are used to determine the target user's satisfaction. Fig. 8 presents percentages of items in the recommendation lists. Facebook (FB) recommends unknown items better than others, because FB usually recommends latest items, but users are usually not interested in them. The percentage of unknown items of traditional method with popularity with

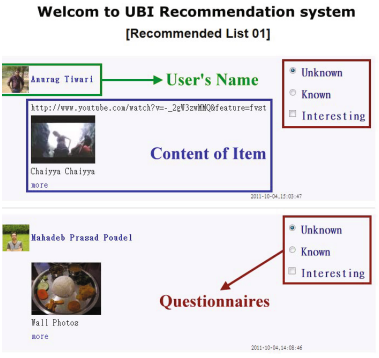


Fig. 7. System user interface for the recommended list

similarity (PS) is worst than that of other methods whereas the percentage of interesting items of traditional method with PS is better than that of FB. Because traditional method recommends items, which are usually already known to users, based on PS among users. Our UBI algorithm can discover unknown items almost same as FB does and interesting items is better than FB and PS. In terms of overall satisfaction with unknown and interesting questionnaires, our algorithm can recommend unknown but interesting items exactly.

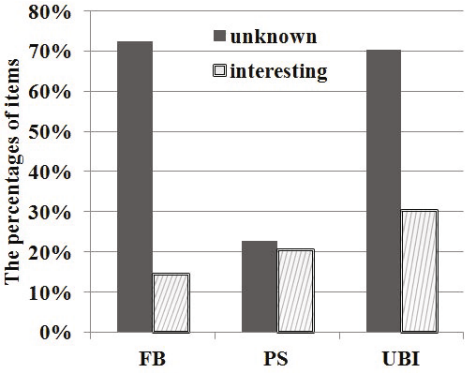


Fig. 8. The percentages of items in the recommendation lists

In addition, we recommend Top-20 items for each recommendation system, then we compared percentages of Top-5 to Top-20. We want to ensure good performance of satisfaction for each stage. Fig. 9 represents the percentage of options checked by users for Top-5 to Top-20. In Fig. 9(a), the UBI algorithm discovers unknown items almost the same as FB, and the percentage of interesting is higher than FB and PS. Besides, Fig. 9(b), (c), and (d) also show this trend.

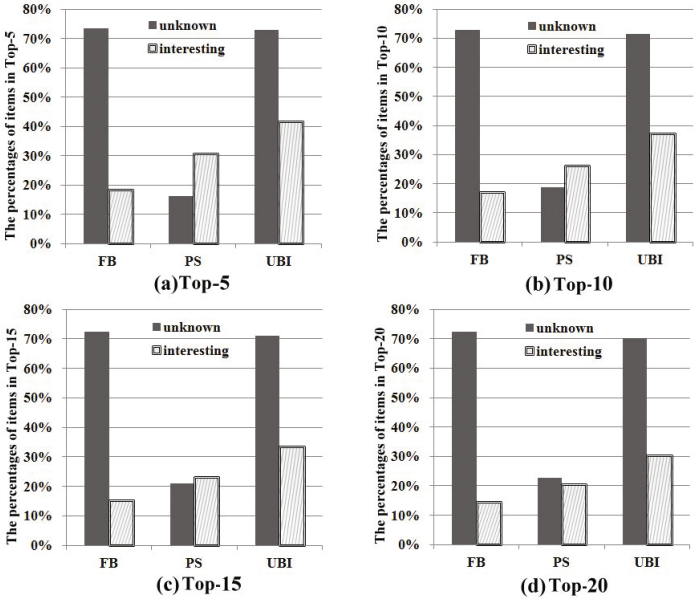


Fig. 9. The percentages of items in the Top-k list

This phenomenon represents not only Top-20 better UBI recommendations, but Top-5 to Top-15 also better than FB and PS. Therefore, we can obtain user's satisfaction from user's feedback is better than that of FB and PS. Finally, We found that our proposed algorithm to discover unknown but interesting items is better than Facebook and the traditional methods.

5 Conclusions

Traditional systems are based on similarity and popularity. This strategy leads to some problems. We proposed an algorithm which recommends unknown but interesting item by utilizing three scores: Quartile-aided Popularity Score, Social Behavior Score, and User Similarity Score. We focus on the communication among users and mutual friends, and discover the unknown but interesting item for user. In other words, we not only consider the similarity but also care about the user's social interaction. Experimental results show that the performance of UBI significantly outperforms that of traditional methods in terms of the percentages of unknown and interesting items in the recommendation list. Our future work could focus on information propagation in social networks, and friend of friend structure and utilize cloud computing techniques to improve the system performance.

References

1. Official facebook blog, http://blog.facebook.com/blog.php?post=15610312130
2. Adamic, L., Adar, E.: Friends and neighbors on the web. Social Networks 25(3), 211–230 (2003)
3. Agrawal, R., Rajagopalan, S., Srikant, R., Xu, Y.: Mining newsgroups using networks arising from social behavior. In: Proceedings of the 12th International Conference on World Wide Web, WWW 2003, pp. 529–535. ACM (2003)
4. Bell, R., Koren, Y., Volinsky, C.: Modeling relationships at multiple scales to improve accuracy of large recommender systems. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2007, pp. 95–104. ACM (2007)
5. Bird, C., Gourley, A., Devanbu, P., Gertz, M., Swaminathan, A.: Mining email social networks. In: Proceedings of the 2006 International Workshop on Mining Software Repositories, MSR 2006, pp. 137–143. ACM (2006)
6. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M.: Combining content-based and collaborative filters in an online newspaper. In: Proceedings of ACM SIGIR Workshop on Recommender Systems (1999)
7. Fu, F., Liu, L., Wang, L.: Empirical analysis of online social networks in the age of web 2.0. Physica A: Statistical Mechanics and its Applications 387(2-3), 675–684 (2008)
8. Geyer, W., Dugan, C., Millen, D.R., Muller, M., Freyne, J.: Recommending topics for self-descriptions in online user profiles. In: Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, pp. 59–66. ACM (2008)
9. Groh, G., Ehmig, C.: Recommendations in taste related domains: collaborative filtering vs. social filtering. In: Proceedings of the 2007 International ACM Conference on Supporting Group Work, GROUP 2007, pp. 127–136. ACM (2007)
10. Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P., Bhattacharjee, B.: Measurement and analysis of online social networks. In: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, IMC 2007, pp. 29–42. ACM (2007)
11. Onuma, K., Tong, H., Faloutsos, C.: Tangent: a novel, 'surprise me', recommendation algorithm. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009, pp. 657–666. ACM (2009)
12. Pazzani, M., Billsus, D.: Content-Based Recommendation Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007)
13. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, WWW 2001, pp. 285–295. ACM (2001)
14. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative Filtering Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 291–324. Springer, Heidelberg (2007)
15. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. In: KDD-2000 Workshop on Text Mining, pp. 109–111 (2000)

The Pattern Next Door: Towards Spatio-sequential Pattern Discovery

Hugo Alatrista Salas[1,3], Sandra Bringay[2], Frédéric Flouvat[3], Nazha Selmaoui-Folcher[3], and Maguelonne Teisseire[1,2]

[1] IRSTEA, UMR TETIS, 500 rue Jean-François Breton, 34093 Montpellier - France
firstname.lastname@teledetection.fr
[2] LIRMM, UMR 5506, 161 rue Ada, 34392 Montpellier - France
firstname.lastname@lirmm.fr
[3] PPME, Université de la Nouvelle-Calédonie, BP R4, Nouméa - New Caledonia
firstname.lastname@univ-nc.nc

Abstract. Health risks management such as epidemics study produces large quantity of spatio-temporal data. The development of new methods able to manage such specific characteristics becomes crucial. To tackle this problem, we define a theoretical framework for extracting spatio-temporal patterns (sequences representing evolution of locations and their neighborhoods over time). Classical frequency support doesn't consider the pattern neighbor neither its evolution over time. We thus propose a new interestingness measure taking into account both spatial and temporal aspects. An algorithm based on pattern-growth approach with efficient successive projections over the database is proposed. Experiments conducted on real datasets highlight the relevance of our method.

1 Introduction

In everyday life, we can observe many phenomena occurring in space and time simultaneously. For example, the movements of a person associate spatial information (e.g. the departure and arrival coordinates) and temporal information (e.g. the departure and arrival dates). Other applications, with more complex dynamics, are much more difficult to analyze. It is the case of spread of infectious disease, which associates spatial and temporal information such as the number of patients, environmental or entomological data. Yuang in [13] describes this concept of dynamics as a *set of dynamic forces impacting the behavior of a system and components, individually and collectively.*

In this paper, we focus on spatio-temporal data mining methods to better understand the dynamics of complex systems for epidemiological surveillance. In the case of dengue epidemics, public health experts know that the evolution of the disease depends on environmental factors (e.g. climate, areas with water points, mangroves...) and interactions between human and vector transmission (e.g. the mosquito that carries the disease). However, the impact of environmental factors and their interactions remain unclear.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 157–168, 2012.

To address these issues, spatio-temporal data mining provides highly relevant solutions through the identification of relationships among variables and events, characterized in space and time without *a priori hypothesis*. For example, in our context, we will discover combinations of changes in environmental factors that lead epidemic peaks in specific spatial configurations. We will show in the related works section that existing methods are not completely adapted to our problem. For this reason, we have define new spatio-sequential patterns, based on an extension of sequential patterns, to link the spatial and temporal dimension. An example of pattern in the dengue context is: *frequently over the past 10 years, if it rains in an area and if there is standing water and high temperatures in the neighborhood, then there is an increase number of mosquitoes in adjacent areas, followed by an increase of dengue cases*. It can be used for analysis by health care professionals, to better understand how environmental factors influence the development of epidemics. Such patterns are very interesting because they enable to capture evolution of areas considering their events and events in adjacent zones. However, they are very difficult to mine because the search space is very large. Proposing scalable methods to find these patterns are consequently very challenging. We have defined an interestingness measure to overcome this problem of scalability and an efficient algorithm based on pattern-growth approach.

In section 2, we review existing spatio-temporal data mining methods and we show that these methods are not suitable for our problem. In section 3, we detail our theoretical framework. In section 4, we present our algorithm called *DFS-S2PMiner*. In section 5, we present experiments on real datasets. The paper ends with our conclusions and future perspectives.

2 Related Work

In this related work section, we are not concerned by the trajectories problematic addressed in [1, 3]. We only focus on methods analyzing the evolution and the interaction of objects or events characteristics through space and time. Early work addressed the spatial and temporal dimensions separately. For example, Han et al. in [4] or Shekhar et al. in [10] looked for spatial patterns or co-location, i.e. subsets of features (object-types) with instances often identified as close in space. In our context, an example of co-location is *within a radius of 200 m, mosquitoes nests are frequently found near ponds*. On the contrary, other authors as Pei et al. in [9] have studied temporal sequences which only take into account the temporal dimension. Tsoukatos et al. in [11] have extended these works to represent sets of environmental features evolving in time. They extract sequences of characteristics that appear frequently in areas, but without taking into account the spatial neighborhood. An example of pattern obtained is: *in many areas, heavy rain occurs before the formation of a pond, followed by the development of mosquito nest*. If these two types of methods, only spatial or temporal, can be very relevant for epidemiological surveillance, they do not capture relations such as: *often, a heavy rain occurs before the formation of a pond followed in a close area by the development of mosquito nests*. In [12], Wang et al. focus on the

extraction of sequences representing the propagation of spatiotemporal events in predefined time windows. They introduce two concepts: *Flow patterns* and *Generalized Spatiotemporal Patterns* in order to extract precisely the sequence of events that occur frequently in some locations. Thus, the authors will be able to identify patterns of the form: *dengue cases appear frequently in area Z1 after the occurrence of high temperatures and the presence of ponds in area Z2.*

However, Huang et al. in [7] found that all the patterns discovered with others approaches are not all the time relevant because they may not be statistically significant and in particular not *"dense"* in space and time. They therefore proposed an interestingness measure taking into account the spatial and temporal aspects to extract global sequence of features. However, they study the events one after another. They don't take into account the interactions such as *often heavy rain and the occurrence of ponds are presented before the development of mosquito nests.* Celik et al. in [2], proposed the concept of *Mixed-Drove Spatiotemporal Co-occurrence Patterns*, i.e. subsets of two or more different event-types whose instances are often located in spatial and temporal proximity (*e.g.* an event-type is *heavy rain* and an instance is *heavy rain in zone Z1 the 10/17/2011*). For similar reasons than Huang, they have proposed a specific monotonic composite interest measure based on spatial and temporal prevalence measures. However, they do not extract the frequent evolutions of event-types over time (events of each instance occur necessarily in the same time slot). For example, we can only extract patterns such as: *heavy rain, ponds and development of mosquito nests are frequently found together in lots of time slots.* Finally, approaches proposed by Wang, Huang and Celik cannot capture the evolution of areas with regard to their set of event-types and the sets of event-types of their neighbors.

In this paper, we describe a method for extracting spatio-temporal sequences of patterns (i.e. sequences of spatial sets of events) called *S2P* (Spatio-Sequential Patterns). We aim at identifying relationships such as: *the presence of dengue cases in an area is often preceded of high temperatures and the presence of water tanks in a neighboring area.* Thus, we will deal with the developments and interactions between the study area and its immediate environment. Moreover, as this type of patterns are very difficult to mine, because of the huge generated search space, we will introduce an interestingness measure to make our approach scalable.

3 Spatio-sequential Patterns: Concepts and Definitions

3.1 Preliminaries

A spatio-temporal database is a structured set of information including geographic components (e.g. neighborhoods, rivers, etc.) and temporal components (e.g. rain, wind). Such a database is defined as a triplet $DB = (D_{\mathrm{T}}, D_{\mathrm{S}}, D_{\mathrm{A}})$ where D_{T} is the temporal dimension, D_{S} the spatial dimension and $D_{\mathrm{A}} = \{D_{\mathrm{A}_1}, D_{\mathrm{A}_2}, \ldots, D_{\mathrm{A}_p}\}$ a set of analysis dimensions associated with attributes. The *temporal dimension* is associated with a domain of values denoted $dom(D_{\mathrm{T}}) = \{T_1, T_2, \ldots, T_t\}$ where $\forall i \in [1..t]$, T_i is a *timestamp* and $T_1 < T_2 < \ldots < T_t$. The

spatial dimension is associated with a domain of values denoted $dom(D_S) = \{Z_1, Z_2, \ldots Z_l\}$ where $\forall i \in [1..l]$, Z_i is a *zone*. We define on $dom(D_S)$ a neighborhood relationship, denoted *Neighbor* by:

$$Neighbor(Z_i, Z_j) = true \text{ if } Z_i \text{ and } Z_j \text{ are neighbors, } false \text{ otherwise} \quad (1)$$

Each dimension D_{A_i} ($\forall i \in [1..p]$) in the set of *analysis dimensions* D_A, is associated with a domain of values denoted $dom(A_i)$. In these domains, the values can be ordered or not.

To illustrate the definitions, we use a sample of weather database, Table 1, which represents weather in three cities on three consecutive days. The table lists temperature (Temp), precipitation (Prec), wind speed (Wind) and gusts in Km/h. The three cities are associated by a neighborhood relationship described in Figure 1.

Table 1. Weather changes in three cities : Z_1, Z_2 et Z_3 on December 22, 23, 24, 2010

City	Date	Temp	Prec	Wind	Gusts
Z_1	12/22/10	T_m	P_m	V_m	-
Z_1	12/23/10	T_m	P_m	V_l	-
Z_1	12/24/10	T_l	P_m	V_m	55
Z_2	12/22/10	T_m	P_m	V_m	-
Z_2	12/23/10	T_l	P_m	V_l	-
Z_2	12/24/10	T_l	P_l	V_m	-
Z_3	12/22/10	T_l	P_m	V_s	75
Z_3	12/23/10	T_m	P_s	V_l	-
Z_3	12/24/10	T_l	P_s	V_s	55

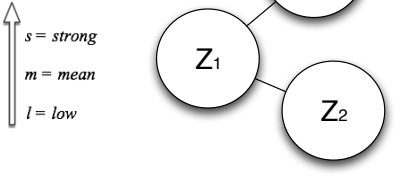


Fig. 1. Neighboring cities

In Table 1, $D_T = \{Date\}$, $D_S = \{City\}$ and $D_A = \{Temp, Prec, Wind, Gusts\}$. The domain of the temporal dimension is $dom(D_T) = \{12/22/10, 12/23/10, 12/24/10\}$ with $12/22/10 < 12/23/10 < 12/24/10$. The domain of spatial dimension is $dom(D_S) = \{Z_1, Z_2, Z_3\}$ with $Neighbor(Z_1, Z_2) = true$, $Neighbor(Z_1, Z_3) = true$ and $Neighbor(Z_2, Z_3) = false$. Finally, for the analysis dimensions *Temp* and *Gusts*, the domains are respectively $dom(Temp) = \{T_m, T_l, T_s\}$ and $dom(Gusts) = \{55, 75\}$.

3.2 Spatio-sequential Patterns

Definition 1. ***Item and Itemset.*** *Let I be an* item*, a literal value for the dimension D_{A_i}, $I \in dom(D_{A_i})$. An* itemset*, $IS = (I_1 I_2 \ldots I_n)$ with $n \leq p$, is a non empty set of* items *such that $\forall i, j \in [1..n], \exists k, k' \in [1..p]$, $I_i \in dom(D_{A_k})$, $I_j \in dom(A_{D_{A'_k}})$ and $k \neq k'$.*

All items in an itemset are associated with different dimensions. An itemset with k items is called k-itemset.

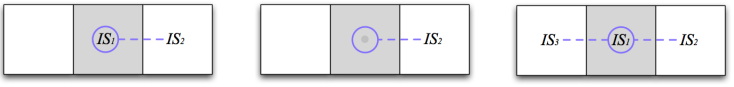
We define the In relationship between *zones* and *itemsets* which describes the occurrence of itemset IS in zone Z at time t in the database DB:

$In(IS, Z, t)$ is true if IS is present in DB for zone Z at time t. In our example, consider the itemset $IS = (T_m P_m V_l)$ then $In(IS, Z_1, 12/23/10)$ is *true* as the itemset $(T_m P_m V_l)$ occurs for zone Z_1 on 12/23/10 (see Table 1).

We now define the notion of *interaction* with neighbor zones.

Definition 2. ***Spatial itemset.*** *Let IS_i and IS_j be two itemsets, we say that IS_i and IS_j are spatially close iff $\exists Z_i, Z_j \in dom(D_S)$, $\exists t \in dom(D_T)$ such that $In(IS_i, Z_i, t) \wedge In(IS_j, Z_j, t) \wedge Neighbor(Z_i, Z_j)$ is true. A pair of itemsets IS_i and IS_j that are spatially close, is called a* ***spatial itemset*** *and denoted by $I_{ST} = IS_i \cdot IS_j$.*

To facilitate notations, we introduce a *n-ary* group operator for itemsets to be assigned by the operator $\cdot$ (*near*), denoted []. The θ symbol represents the *absence* of itemsets in a zone. Figure 2 shows the three types of spatial itemsets that we can build with the proposed notations. The dotted lines represent the spatial dynamics.



(a) (b) (c)

Fig. 2. Graphical representation of spatial itemsets (a) $IS_1 \cdot IS_2$ (b) $\theta \cdot IS_2$ (c) $IS_1 \cdot [IS_2; IS_3]$

The spatial itemset $I_{ST} = (T_m \cdot (V_l P_m))$ describes that events T_m and $V_l P_m$ occur in neighboring zones at the same time. The spatial itemset $I_{ST} = (\theta \cdot [T_m; P_l])$ indicates that T_m and P_l occur in two different zones neighbor to a zone where no event appears.

Definition 3. ***Inclusion of spatial itemset.*** *A spatial itemset $I_{ST} = IS_i \cdot IS_j$ is* included*, denoted $\subseteq$, in another spatial itemset $I'_{ST} = IS'_k \cdot IS'_l$, iff $IS_i \subseteq IS'_k$ and $IS_j \subseteq IS'_l$.*

The spatial itemset $I_{ST} = (T_m P_m \cdot V_l)$ is *included* in the spatial itemset $I'_{ST} = (T_m P_m \cdot V_l 55)$ because $(T_m P_m) \subseteq (T_m P_m)$ and $(V_l) \subseteq (V_l, 55)$.

We now define the notion of zones *evolution* according to their spatial neighborhood relationship.

Definition 4. ***Spatial Sequence.*** *A spatial sequence or simply* ***S2*** *is an ordered list of spatial itemsets, denoted $s = \langle I_{ST_1} I_{ST_2} \ldots I_{ST_m} \rangle$ where I_{ST_i}, $I_{ST_{i+1}}$ satisfy the constraint of temporal sequentiality for all $i \in [1..m-1]$.*

A S2 $s = \langle (T_m)(\theta \cdot [P_l; V_s])(V_l \cdot [P_l; T_l]) \rangle$ is illustrated in figure 3 for the zone $Z1$, where the arrows represent the temporal dynamics and the dotted lines represent the environment.

A relationship generalization (or specialization) between S2's is defined as follows:

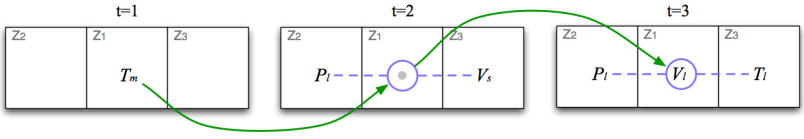


Fig. 3. Example of the spatio-temporal dynamic

Definition 5. ***Inclusion of S2.*** *A S2* $s = \langle I_{ST_1} I_{ST_2} \dots I_{ST_m} \rangle$ *is more specific than a S2* $s' = \langle I'_{ST_1} I'_{ST_2} \dots I'_{ST_n} \rangle$*, denoted* $s \preceq s'$*, if there exists* $j_1 \leq \dots \leq j_m$ *such that* $I_{ST_1} \subseteq I'_{ST_{j_1}}, I_{ST_2} \subseteq I'_{ST_{j_2}}, \dots, I_{ST_m} \subseteq I'_{ST_{j_m}}$.

A S2 $s = \langle (T_l P_m \cdot P_l V_s)(55) \rangle$ is included in the S2 $s' = \langle (T_l P_m \cdot P_l V_s)(55 \cdot V_s) \rangle$ because $(T_l P_m \cdot P_l V_s) \subseteq (T_l P_m \cdot P_l V_s)$ and $(55) \subseteq (55 \cdot V_s)$.

For a specific zone, we note s_Z the associated spatial data sequence in the database DB. s_Z contains or supports a spatial sequence s if s is a subsequence of s_Z. The support of a spatial sequence s is thus defined as the number of zone supporting s. If the support of the spatial sequence is greater than a user-defined threshold, the sequence is frequent and corresponds to a **spatio-sequential pattern** ($S2P$). Nevertheless, in a spatio-temporal context, we need to define a more precise and suitable prevalence measure, as explained in the next section.

3.3 Spatio-temporal Participation

The proposed spatio-sequential pattern allow to tackle both spatial and temporal issues. In order to manage in an efficient way the mining of such patterns, a new filtering measure has to be defined. To highlight the participation of an item in a spatial sequence, we propose an adaptation of the participation index [6] which is a combination of two measures: **spatial participation index** and **temporal participation index** taking into account respectively the spatial dimension and the number of occurrences in time.

Definition 6. ***Spatial participation ratio*** *Let* s *be a spatial sequence and* I *be an item of* s*, the* spatial participation ratio *for* I *in* s*, denoted by* $SPr(s, I)$ *is the number of zones which contain* s *divided by the number of zones where the item* I *appears in the whole database:*
$SPr(s, I) = \frac{Supp(s)}{Supp(I)}$

Definition 7. ***Spatial participation index*** *Let* $s = \langle I_{ST_1}, I_{ST_2}, \dots I_{ST_n} \rangle$ *be a spatial sequence, the* spatial participation index *of* s *denoted* $SPi(s)$ *is the minimum of* spatial participation ratio*:*
$SPi(s) = MIN_{\forall I \in dom(A), I \in s} \{SPr(s, I)\}$

Definition 8. ***Temporal participation ratio*** *Let* s *be a spatial sequence and* I *be an item of* s*, the* temporal participation ratio *for* I *in* s *denoted* $TPr(s, I)$ *is the number of occurrences of* s *(i.e. the number of instances over time) divided by the total number of occurrences of* I*:*
$TPr(s, I) = \frac{NbOccurrences(s)}{NbOccurrences(I)}$

Definition 9. ***Temporal participation index*** *Let* $s = \langle I_{ST_1}, I_{ST_2}, \ldots I_{ST_n} \rangle$ *be a spatial sequence, the* temporal participation index *of* s *denoted* $TPi(s)$ *is the minimum of* temporal participation ratio*:*
$TPi(s) = MIN_{\forall I \in dom(A_i), I \in s} \{TPr(I, s)\}$

We define the **spatio-temporal participation index** of a spatial sequence s, $STPi(s)$, as:

$$STPi(s) = 2 * \frac{SPi(s) * TPi(s)}{SPi(s) + TPi(s)} \tag{2}$$

Given a spatio-temporal database DB, the problem of spatio-sequential pattern mining is to find all spatial sequences whose spatio-temporal participation index is greater than a user-specified threshold *min_stpi*.

Note that the predicate "STPi is greater than a user-threshold" is antimonotonic. If a spatio-sequential pattern s is not frequent, all patterns s' such as s is included in s' ($s \preceq s'$), are also not frequent. This property is used in our pattern mining algorithm to prune the search space and quickly find frequent spatio-sequential patterns.

4 Extraction of Spatio-sequential Patterns

In this section, we propose an algorithm called *DFS-S2PMiner* to extract spatio-sequential patterns considering both spatial and temporal aspects. DFS-S2PMiner adopts a depth-first-search strategy based on successive projections of the database such as FP-Growth [5] and Prefixspan [8] for scalability purpose. Specifically, this algorithm is based on the *pattern-growth* strategy used in [5]. The principle of this approach is to extract frequent patterns without a candidate generation step. This approach recursively creates a projected database, associates it with a fragment of frequent pattern, and "mines" each projected database separately. The frequent patterns are extended progressively along a depth-first exploration of the search space.

First, we introduce the definition of the projection of a spatio-temporal database used in the algorithm. Let s be a spatio-sequence of the database DB. The projection of database DB w.r.t. s, denoted $DB|_s$, is the set of suffixes of s in DB.

The algorithm 1 describes our recursive algorithm DFS-S2PMiner. First, the set of frequent items I and $\theta \cdot I$, denoted F_1, is extracted from the projected database $DB|_\alpha$ (line 1 of Algorithm 1). These items constitute extensions of sequence α. Note that in the first recursive call, $DB|_\alpha$ corresponds to the initial database DB (since $\alpha = \{\}$). Then, for each of these items $X \in F_1$, we extend the spatio-sequential pattern α with X (lines 3 and 4). Two types of extension are possible : 1) adding X to the last spatial itemset of the sequence α (line 3) or 2) inserting X after (i.e. the next time) the last spatial itemset of α (line 4). We check the measure of interest for these two spatio-sequential patterns (lines 5 and 9) and record frequent ones in the set of solutions F (lines 6 and 10). For each frequent pattern, the algorithm then performs another projection of the

database using $DB|_\alpha$ and recursively extends the pattern by invoking again the algorithm (lines 7 and 11). The algorithm stops when no more projections can be generated.

Algorithm 1. DFS-S2PMiner

– **Main routine**

Require: A spatio-temporal database DB and a user-defined threshold min_stpi
Ensure: A set of frequent spatio-sequential patterns F
$\alpha \leftarrow \{\}$
Call $Prefix\text{-}growthST(\alpha, min\_stpi, DB|_\alpha, F)$

– **Prefix-growthST** $(\alpha, min\_stpi, DB|_\alpha, F)$

Require: a spatio-sequential pattern α , the user-defined threshold min_stpi, the projection $DB|_\alpha$ of the spatio-temporal database on α, and F a set of frequent spatio-temporal patterns;
1. $F_1 \leftarrow$ {a set of frequents items I and $\theta \cdot I$ on $DB|_\alpha$, with $I \in \bigcup_{i \in [1..p]} dom(D_{A_i})$ }
2. **for all** $X \in F_1$ **do**
3. $\beta \leftarrow \alpha X$
4. $\delta \leftarrow \alpha(X)$
5. **if** $STPi(\beta) \geq min\_stpi$ **then**
6. $F \leftarrow F \cup \beta$;
7. Prefix-growthST$(\beta, min_stpi, DB|_\beta, F)$
8. **end if**
9. **if** $STPi(\delta) \geq min\_stpi$ **then**
10. $F \leftarrow F \cup \delta$;
11. Prefix-growthST$(\delta, min_stpi, DB|_\delta, F)$
12. **end if**
13. **end for**

We use our running example (Table 1 and Figure 1) with $min\_stpi = 2/3$ to illustrate this algorithm.

Iteration 1 $(\alpha = \{\})$

– ***Extraction on frequent items and spatial items (line 1).*** The first step is to extract frequent items and frequent spatial items from DB, let:

$$F_1 = \{P_m : 3, T_m : 3, V_m : 2, V_l : 3, T_l : 3, 55 : 2, \theta \cdot T_m : 3,$$
$$\theta \cdot P_m : 3, \theta \cdot V_m : 3, \theta \cdot V_l : 3, \theta \cdot T_l : 3, \theta \cdot 55 : 3\}$$

– ***Extension of current sequence*** α ***(lines 3-4).***
– ***STPi processing and Recording solutions (lines 5-6 and 9-10).***
– ***Projection and Recursive call (lines 7 and 11).*** For each frequent item I and $\theta \cdot I$, the algorithm calculates the corresponding projection of the database. For example, for the frequent item P_m, we obtain the following projection (see Table 2). Each of these projected database is used in a recursive call to find its frequent super-sequences.

Iteration 2 $(\alpha = \langle (P_m) \rangle)$

– ***Extraction on frequent items and spatial items (line 1).*** The first recursive call will build the super-sequences with the prefix $\langle (P_m) \rangle$ from the projected database of Table 2. Specifically, the algorithm will find frequent

Table 2. Projected database of $\langle(P_m)\rangle$

Zones	Sequences	Neighbors	Neighbor sequences
Z_1	$S_1 = \langle(\_\boldsymbol{V_m})(T_m\boldsymbol{P_m}V_l)(T_l\boldsymbol{P_m}\boldsymbol{V_m}55)\rangle$	Z_2	$S_2 = \langle(\_V_m)(T_lP_mV_l)(T_lP_lV_m)\rangle$
		Z_3	$S_3 = \langle(\_V_s75)(T_mP_sV_l)(T_lP_sV_s55)\rangle$
Z_2	$S_2 = \langle(\_\boldsymbol{V_m})(T_l\boldsymbol{P_m}V_l)(T_lP_l\boldsymbol{V_m})\rangle$	Z_1	$S_1 = \langle(\_V_m)(T_mP_mV_l)(T_lP_mV_m55)\rangle$
Z_3	$S_3 = \langle(\_V_s75)(T_mP_sV_l)(T_lP_sV_s55)\rangle$	Z_1	$S_1 = \langle(\_V_m)(T_mP_mV_l)(T_lP_mV_m55)\rangle$

items in the projected database (line 1) and extend $\langle(P_m)\rangle$ (line 2 - 4). The frequent items obtained from $DB|_{\langle(P_m)\rangle}$ are: $\{V_m : 2, T_m : 2, P_m : 2, V_l : 3, T_l : 3, 55 : 2, \theta \cdot V_m : 3, \theta \cdot T_l : 3, \theta \cdot P_m : 3, \theta \cdot V_l : 3, \theta \cdot T_m : 3, \theta \cdot 55 : 3\}$

- ***Extension of current sequence*** α ***(lines 3-4).*** The first frequent item found is $\langle V_m\rangle : 2$. Therefore, we can build two spatial sequences: $\langle(P_mV_m)\rangle$ (line 3) and $\langle(P_m)(V_m)\rangle$ (line 4).
- ***STPi processing and Recording solutions (lines 5-6 and 9-10).*** The spatio-sequential pattern $\langle(P_m)(V_m)\rangle$ with $STPi = 2/3$ is frequent (line 9).
- ***Projection and Recursive call (lines 7 and 11).*** Thus, the algorithm uses this pattern to make a new projection (see Table 3) and to recursively search all frequent super-sequences with the prefix $\langle(P_m)(V_m)\rangle$.

Table 3. Projected database of $\langle(P_m)(V_m)\rangle$

Zones	Sequences	Neighbors	Neighbor sequences
Z_1	$S_1 = \langle(T_mP_mV_l)(T_lP_mV_m55)\rangle$	Z_2	$S_2 = \langle(T_l\boldsymbol{P_m}V_l)(T_l\boldsymbol{P_m}V_m)\rangle$
		Z_3	$S_3 = -$
Z_2	$S_2 = \langle(T_lP_mV_l)(T_lP_lV_m)\rangle$	Z_1	$S_1 = \langle(T_m\boldsymbol{P_m}V_l)(T_l\boldsymbol{P_m}V_m55)\rangle$
Z_3	$S_3 = \emptyset$	Z_1	$S_1 = \langle(T_m\boldsymbol{P_m}V_l)(T_l\boldsymbol{P_m}V_m55)\rangle$

Iteration 3 ($\alpha = \langle(P_mV_m)\rangle$)

- ***Extraction on frequent items and spatial items (line 1).*** The frequent items obtained for $DB|_{\langle(P_mV_m)\rangle}$ are: $\{V_m : 2, P_m : 2, V_l : 2, T_l : 2, \theta \cdot V_m : 3, \theta \cdot T_l : 3, \theta \cdot P_m : 3, \theta \cdot V_l : 3, \theta \cdot 55 : 3\}$.
- ***Extension of current sequence*** α ***(lines 3-4).*** For example, the spatial item $\theta \cdot P_m : 3$ is one of the frequent items. In this case, the algorithm builds the spatio-sequential pattern $\langle(P_m)(V_m)(\theta \cdot P_m)\rangle$.
- ***STPi processing and Recording solutions (lines 5-6 and 9-10).*** This pattern is frequent with a $STPi = 1$ because $\langle\theta \cdot P_m\rangle$ appears in all times and zones (see Table 3).
- ***Projection and Recursive call (lines 7 and 11).*** When all frequent items are projected, the algorithm goes through another *branch* of the search space, i.e. patterns beginning with $\langle(T_m)\rangle$ (see set F_1)

The algorithm thus proceeds generally in the same way whether items are spatial or not. The main difference is how to compute the support. The support of a spatial item is the number of zones where the item occurs at least once in

their neighborhood (so we have $\theta \cdot V_l : 3$ in Table 2). Notice that when the algorithm extends a pattern of type $\langle (I_{ST_1})(I_{ST_2})...(I_{ST_k} \cdot X) \rangle$ with a common item $\theta \cdot Y$, the operator of *n-ary* group is used to represent the sequence as $\langle (I_{ST_1})(I_{ST_2}) \ldots (I_{ST_k} \cdot [X;Y]) \rangle$.

5 Experiments

The approach proposed in this paper has been integrated in a Java prototype, and it has been experimented on two real datasets. The first one represents the evolution of dengue infection in a city during an epidemic (26 dates). The city is divided in 81 districts each one characterized by 12 epidemic and environmental attributes (e.g. number of dengue cases, precipitation per day or presence of pools). The second dataset is a record of biological indicators in the Saône rivers, for example, IBGN (Standardized Global Biological Index) and IBD (Biological Diatom Index). These indicators are associated with hydrological stations along the watercourse and raised up made by some stations along the watersheds of the Saône. This dataset includes 815 samples associated to 223 stations (zones) and 10 attributes.

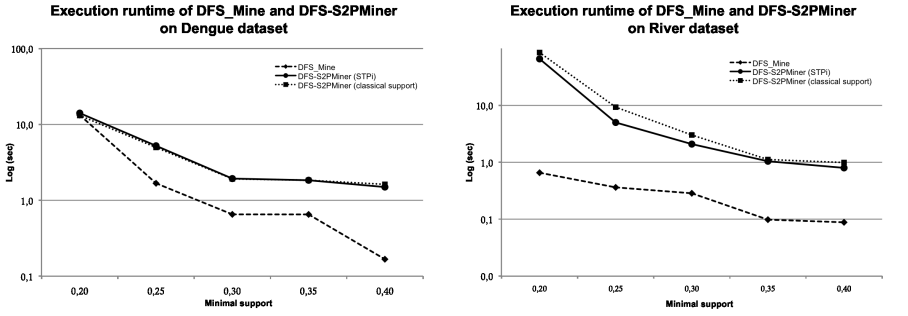
We compared our approach with the work proposed by Tsoukatos [11] since it is the closest work. Indeed, this work extracts sequences of itemsets representing the evolution of each zone individually (but without taking into account neighbors as in our approach). Experiments have been done on an Intel Core I5 processor with 4G of RAM on Linux.

First, a qualitative evaluation of the results was done. We compared the patterns obtained by our approach with the ones obtained by the DFS_Mine algorithm of Tsoukatos on the dengue dataset.

For example, both approaches could find classical sequential patterns such as *"few pools, few precipitations and few graveyard are followed by few dengue, few precipitations and wind"*. However, our approach could also find complex patterns such as *"few pools, few precipitations and few graveyard, followed by few pools and few precipitations in neighbor zones, are followed by few dengue in neighbor zones"*. This example gives an idea of the richness of our patterns by enabling to highlight the influence of neighbor areas.

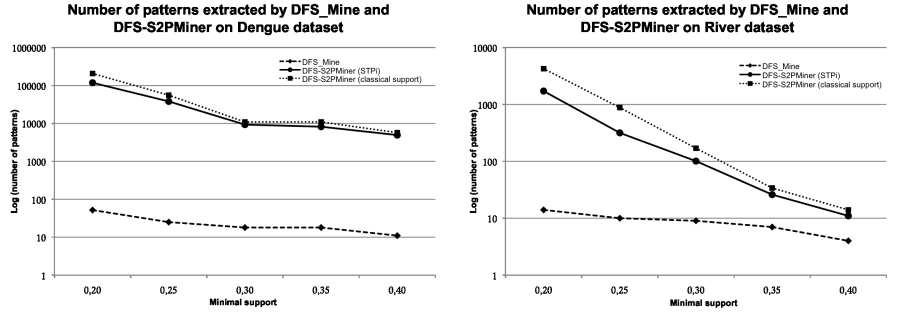
When using the spatio-temporal participation index as measure of interest, we can't compare any further the extracted patterns since prevalence measures are different. While the approach of Tsoukatos keeps sequences occurring in many zones but not necessarily several times, our approach keeps sequences occurring in many zones and several times. The interest of our proposal is to consider the temporal weight of patterns.

Second, a quantitative evaluation of our approach was done. We compared the execution time of our algorithm with the DFS_Mine algorithm proposed by Tsoukatos in [11]. Figure 4 shows execution times of DFS_Mine (classical support) and DFS-S2PMiner algorithms (using classical support and spatio-temporal participation index) on the studied datasets for several thresholds. Execution times are relatively similar while our approach is doing more complex



(a) (b)

Fig. 4. Execution runtime of DFS_Mine and DFS-S2PMiner algorithms on (a) Dengue dataset (b) River dataset



(a) (b)

Fig. 5. Number of patters extracted by DFS_Mine and DFS-S2PMiner algorithms on (a) Dengue dataset (b) River dataset

processing. Indeed, as shown by figure 5, the STPi measure allows an efficient pruning of the search space, even for the large dataset of the Saône river.

6 Conclusion and Perspectives

In this paper, we propose a new concept of spatio-temporal patterns called spatio-sequential patterns *(S2P)*. This concept enables to analyze the evolution of areas considering their set of features and their neighboring environment. An example application of these patterns is the study of the spatiotemporal spread of dengue w.r.t. epidemic, district and environmental data. A formal framework is established to define *S2P* generically. To extract these patterns, we propose a generic method called *DFS-S2PMiner* based on a depth-first strategy. A new prevalence measure has been defined to cope with the limits of the classical support w.r.t. spatial and temporal aspects. Our proposal has been experimented on two real datasets. Results show the interest of the approach to extract efficiently rich spatio-temporal patterns.

Among possible future developments, we plan to extend the concept of neighborhood to n neighborhoods while allowing scalability. No new definitions are needed but an heuristic exploration of the search space may be required.

Acknowledgments. We wish to thank the Department of Health and Social Affairs of New Caledonia, The Institute Pasteur, IRD and UNC for giving us the `Dengue` data set (Convention 2010). This work was partly funded by French contract ANR-2010-COSI-012 FOSTER.

References

[1] Cao, H., Mamoulis, N., Cheung, D.: Mining frequent spatio-temporal sequential patterns. In: Proc. of IEEE ICDM, pp. 82–89 (2005)
[2] Celik, M., Shekhar, S., Rogers, J., Shine, J.: Mixed-drove spatiotemporal co-occurrence pattern mining. Proc. of IEEE TKDE 20(10), 1322–1335 (2008)
[3] Giannotti, F., Nanni, M., Pinelli, F., Pedreschi, D.: Trajectory pattern mining. In: Proc. of ACM SIGKDD, pp. 330–339 (2007)
[4] Han, J., Koperski, K., Stefanovic, N.: Geominer: a system prototype for spatial data mining. In: Proc. of ACM SIGMOD, SIGMOD 1997, pp. 553–556 (1997)
[5] Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., Hsu, M.-C.: Freespan: frequent pattern-projected sequential pattern mining. In: Proc. of ACM SIGKDD, KDD 2000, pp. 355–359 (2000)
[6] Huang, Y., Shekhar, S., Xiong, H.: Discovering colocation patterns from spatial data sets: a general approach. Proc. of IEEE TKDE 16(12), 1472–1485 (2004)
[7] Huang, Y., Zhang, L., Zhang, P.: A framework for mining sequential patterns from spatio-temporal event data sets. Proc. of IEEE TKDE 20(4), 433–448 (2008)
[8] Mortazavi-Asl, B., Pinto, H., Dayal, U.: PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth. In: Proc. of 17th International Conference on Data Engineering, pp. 215–224 (2000)
[9] Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., Hsu, M.-C.: Mining sequential patterns by pattern-growth: The prefixspan approach. Proc. of IEEE TKDE 16(11), 1424–1440 (2004)
[10] Shekhar, S., Huang, Y.: Discovering Spatial Co-location Patterns: A Summary of Results. In: Jensen, C.S., Schneider, M., Seeger, B., Tsotras, V.J. (eds.) SSTD 2001. LNCS, vol. 2121, pp. 236–256. Springer, Heidelberg (2001)
[11] Tsoukatos, I., Gunopulos, D.: Efficient Mining of Spatiotemporal Patterns. In: Jensen, C.S., Schneider, M., Seeger, B., Tsotras, V.J. (eds.) SSTD 2001. LNCS, vol. 2121, pp. 425–442. Springer, Heidelberg (2001)
[12] Wang, J., Hsu, W., Li Lee, M.: Mining Generalized Spatio-Temporal Patterns. In: Zhou, L.-z., Ooi, B.-C., Meng, X. (eds.) DASFAA 2005. LNCS, vol. 3453, pp. 649–661. Springer, Heidelberg (2005)
[13] Yuan, M.: Geographic Data Mining and Knowledge Discovery, 2nd edn., pp. 347–365

Accelerating Outlier Detection with Uncertain Data Using Graphics Processors

Takazumi Matsumoto and Edward Hung

Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong
{cstmatsumoto,csehung}@comp.polyu.edu.hk

Abstract. Outlier detection (also known as anomaly detection) is a common data mining task in which data points that lie outside expected patterns in a given dataset are identified. This is useful in areas such as fault detection, intrusion detection and in pre-processing before further analysis. There are many approaches already in use for outlier detection, typically adapting other existing data mining techniques such as cluster analysis, neural networks and classification methods such as Support Vector Machines. However, in many cases data from sources such as sensor networks can be better represented with an uncertain model. Detecting outliers with uncertain data involves far more computation as each data object is usually represented by a number of probability density functions (*pdf*s).

In this paper, we demonstrate an implementation of outlier detection with uncertain objects based on an existing density sampling method that we have parallelized using the cross-platform OpenCL framework. While the density sampling method is a well understood and relatively straightforward outlier detection technique, its application to uncertain data results in a much higher computational workload. Our optimized implementation uses an inexpensive GPU (Graphics Processing Unit) to greatly reduce the running time. This improvement in performance may be leveraged when attempting to detect outliers with uncertain data in time sensitive situations such as when responding to sensor failure or network intrusion.

1 Introduction

In recent years there has been increased interest in mining uncertain data [1]. A significant amount of data collected, such as from temperature sensors, contain some degree of uncertainty, as well as possibly erroneous and/or missing values. Some statistical techniques such as privacy-preserving data mining may deliberately add uncertainty to data. In addition, with the proliferation of affordable, capacious storage solutions and high speed networks, the quantity of data collected has increased dramatically. To quickly deal with the large quantities of mostly uninteresting data, outlier detection is a useful technique that can be used to detect interesting events outside of typical patterns. However, uncertainty adds greatly to the complexity of finding outliers as uncertain objects are not represented by a single point, but rather a probabilistic object (i.e. the point could be anywhere in the given space with some probability). This increase in complexity leads to the problem of reduced scalability of algorithms to larger amounts of data.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 169–180, 2012.

Within a similar time frame, multi-core processors and most recently general purpose computing using graphics processors (GPGPU) have become popular, cost effective approaches to provide high performance parallel computing resources. Modern GPUs are massively parallel floating point processors attached to dedicated high speed memory – for a fraction of the cost of traditional highly parallel processing computers. Programming frameworks such as NVIDIA CUDA and OpenCL now allow for programs to take advantage of this previously underutilized parallel processing potential in ordinary PCs and accelerate computationally intensive tasks beyond typical applications in 3D graphics, seeing use in scientific, professional and home applications (such as video encoding).

Our contributions in this paper are a modified density sampling algorithm and implementations for fast parallel outlier detection with uncertain data using this parallel processing resource. Our implementation has been optimized for the features and restrictions of the OpenCL framework and current GPUs.

This paper is organized as follows: Sect. 2 briefly covers related work in the field of outlier detection with uncertain data as well as other GPU accelerated outlier detection techniques. Sect. 3 describes our modified algorithm used in this paper for outlier detection with uncertain data. Sect. 4 details our implementation of the algorithm, with attention to key points in parallelization and optimization using the OpenCL framework. Sect. 5 describes our testing methodology and demonstrates the effectiveness of our OpenCL-based approach in greatly improving performance using both GPU and CPU hardware. Sect. 6 summarizes our contributions and concludes this paper.

2 Related Work

Outlier detection is a well established and commonly used technique for detecting data points that lie outside of expected patterns. The prototypical approach to outlier detection is as a by-product of clustering algorithms [2]. In a clustering context, an algorithm such as DBSCAN [3] will exclude data points that are not close (given an appropriate metric such as distance or density) to other objects. Later, more approaches were proposed for outlier detection, such as Local Outlier Factor (based on k-nearest-neighbors), Support Vector Machines, and neural networks.

Data mining applications such as outlier detection are also candidates for parallelization to reduce running time [4] as in typical cases there is a large amount of data that is processed by a small number of routines, possibly in real-time or interactively (for example, in an intrusion detection system). These tasks are said to be ‘data parallel’, and such tasks are well suited for execution on a GPU [2]. Unlike conventional parallel processing computers that have many complex CPU cores, a modern GPU consists of a large number of simple ‘stream processors’ that are individually capable of only a few operations. However, the ability to pack many stream processors into the same space as a single CPU core gives GPUs a large advantage in parallelism. A similarly parallel traditional CPU-based system would be significantly more costly and complex.

The two most popular programming frameworks for GPGPU are C for CUDA, a proprietary solution developed by NVIDIA Corporation, and OpenCL, an open standard backed by multiple companies including Intel, AMD, NVIDIA and Apple. With both

CUDA and OpenCL, work is split from the host (i.e. the CPU) to kernels that execute on a computing device (typically GPUs). Kernels contain the computationally intensive tasks, while the host is tasked with managing the other computing devices. A single kernel can be executed in parallel by many worker threads on a GPU.

Several outlier detection algorithms [2] [5] [6] [7] have been adapted for acceleration with GPUs using CUDA and have seen significant reductions in running times (e.g. a hundred fold improvement in [2]). In this paper, we opt to use OpenCL, as it provides a high degree of portability between different manufacturers of GPUs, as well as the ability to execute the same parallel code on a CPU for comparison.

While there is already a large body of work in the area of accelerating outlier detection on regular (certain) data, often in real world cases data collected has some degree of uncertainty or error [1], for instance, a network of temperature sensors monitoring a greenhouse. Moreover, some statistical techniques such as forecasting and privacy preserving data mining will naturally be uncertain. These uncertainties can be represented by a number of common probability density functions (e.g. Gaussian distribution), which offer a convenient closed form representation. However, sampling a *pdf* for outlier detection using a typical distance or density based approach will result in greatly increased running time due to the computational load from calculations from all the sampled points (e.g. LOF has a complexity of $O(n^2)$ [2], where n would in this case be the total number of samples).

The work in [8] introduces outlier detection on uncertain data as records in a database, with each record having a number of attributes (dimensions). Each dimension has a *pdf*, and the objective is to find data points in an area with data density η (expressed as the η-probability of a data point) less than a threshold value δ, i.e. a (δ, η)-outlier. As it is assumed that outliers have low density in some subspace of the data, each subspace is explored and in each subspace and outliers are removed.

It is noted [8] that it is impractical to determine η-probabilities directly, so a sampling approach of the *pdf*s is used. As the sampling process is a very time consuming operation, [8] also proposes a 'microclustering' technique to reduce the number data objects into clusters. However, in this paper we do not explore microclustering of the data to focus on the performance of density sampling on a GPU.

3 Algorithm for Outlier Detection with Uncertain Data

In this paper, we propose modification of the density sampling algorithm proposed in [8] to optimize it for GPU acceleration. We will first recap the outlier detection approach and terminology.

Let dataset $\mathcal{D}$ contain n uncertain data objects, with each object d_i having m dimensions. For each object, each dimension has a *pdf* that is assumed to be independent. Each object is represented by its mean value $\bar{X}_i$. The *pdf* for $\bar{X}_i$ along dimension j is denoted by $h_i^j(\cdot)$ and the standard deviation of $h_i^j(\cdot)$ is denoted $\psi_j(\bar{X}_i)$. In the case of data stored in certain form (i.e. without standard deviation), uncertainty can be estimated from the calculated standard deviation of each dimension using the Silverman approximation suggested in [8].

It is defined that the η-probability of object $\bar{X}_i$ is the probability that $\bar{X}_i$ lies in a subspace with overall data density of at least η. A subspace is defined as the objects in a subset of the m dimensions, while overall data density is defined by *pdf*s of each object. The probability p_i of $\bar{X}_i$ in a subspace of dimensionality r with overall data density of at least η can be found by solving the following integral:

$$p_i = \int_{h(x_1,\ldots,x_r)} \prod_{j=1}^{r} h_i^j(x_j)\, dx_j \tag{1}$$

Note that $h(x_1, \ldots, x_r)$ represents the overall probability density function on all coordinates in the given subspace. However, as p_i is difficult to calculate precisely with Eq. 1, it can be estimated using a density sampling algorithm, using s samples:

EstimateProbability(d_i, η, r, s)
Let $F_i^j(\cdot)$ be the inverse cumulative distribution function of *pdf* $h_i^j(\cdot)$
$success = 0$, $runs = 0$
for s times **do**
  for $j = 1$ to r **do**
    $y =$ a uniform random value $[0, 1]$
    for all d_k in $\mathcal{D}$ **do**
      $density_i = density_i + h_k^j(F_i^j(y))$
    end for
    $density_i = density_i / |\mathcal{D}|$
  end for
  if $density_i > \eta$ **then**
    $success = success + 1$
  end if
  $runs = runs + 1$
end for
return $(success/runs)$

By sampling an object at multiple random points, calculating the overall data density at each sampled point, and counting how many of those sampled points exceed η, the probability that object lies in a subspace with data density of at least η (i.e. η-probability) can be estimated. It is also evident that the requirement to calculate overall data density at each sampled point presents a large computational workload.

Finally, object $\bar{X}_i$ is defined as a (δ, η)-outlier if the η-probability of $\bar{X}_i$ in any subspace is less than a user parameter δ, that is, the object has a low probability of lying in a region of high data density.

The overall algorithm is presented in pseudo code form as follows:

DetectOutlier($\mathcal{D}$, η, δ, r, s)
$\mathcal{O} = null$
$i = 1$
$\mathcal{C}_i = \{$ First dimension of data points in $\mathcal{D}$ $\}$
while $\mathcal{C}_i$ is not empty and $i \leq r$ **do**
  For each object $d \in \mathcal{C}_i - \mathcal{O}$ calculate *EstimateProbability*(d, i, s, η)
  Add any objects with η-probabilities $< \delta$ to $\mathcal{O}$

$\mathcal{C}_{i+1}$ = for each point in $\mathcal{C}_i - \mathcal{O}$ append corresponding dimension $i+1$ from $\mathcal{D}$
$i = i + 1$
end while

Note that the outlier detection algorithm uses a roll-up approach [8], starting with a one dimensional subspace and adding more dimensions for each iteration. Each subspace is tested for outliers and any outliers are discarded from further consideration in other subspaces.

4 Serial and Parallel Implementations

In order to compare performance, we implemented the algorithm described in the previous section in two ways: a traditional serial implementation in C++ (for the CPU) and a parallel implementation optimized for the OpenCL framework (for both CPU and GPU). In this section, we detail the key points to our serial and parallel implementations.

Note that in this paper, we assume all dimensions and their *pdf*s are independent of each other. Density at a single point in a given data object is estimated by taking the *pdf* of all dimensions of all data objects. This process is repeated for each sample of each data object. In this case, the *pdf* is given by the Gaussian function $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$. The calculation of the inverse cumulative distribution function (inverse *cdf*) is more complicated due to the absence of a closed form representation. In this implementation, it is calculated numerically using the technique described in [9].

4.1 Serial Methods

Within the serial implementation are two methods referred to as 'iterative' and 'single pass'. As noted previously, the density sampling algorithm uses a roll-up approach to outlier detection starting with a single dimension and adding more dimensions for each subsequent iteration. Since each dimension is considered independent, every combination of subspaces does not need to be considered without loss of generality. Any outliers found in a given subspace are excluded from later subspaces. The 'iterative' method uses this roll-up approach as described in Sect. 3. In contrast, the simpler 'single pass' method only tests the entire problem space, that is, rather than looping through subspaces of dimensionality 1 to r in *DetectOutlier*, one subspace of dimensionality r is tested. This effectively averaging out all the densities in each dimension. As shown in Sect. 5, this has a marked impact on performance (running time) as well as quality.

4.2 Parallel Methods

The parallel OpenCL implementation follows a similar path, with two methods referred to as 'early reject' and 'no early reject'. As described in Sect. 3, 'early reject' generates comparable results to the serial iterative method, and 'no early reject' generates comparable results to the serial single pass approach. However, the OpenCL implementation differs in some key ways to the relatively straightforward serial implementation.

When a kernel executes on a computing device such as a GPU, it should take into account a different architecture to a regular CPU. To better leverage the GPU using

OpenCL, this implementation uses several optimizations such as the use of single precision floating point values and special hardware accelerated mathematical functions (*native* functions). Single precision floating point values are used extensively in graphics, and thus GPUs are optimized for many single precision functions. By avoiding double precision there are significant performance improvements at the cost of a small amount of quality. However this is dependent on the hardware platform, and on our test platform the CPU offered worse performance running OpenCL code with these optimizations. As such, for fairness the CPU OpenCL implementation uses a simpler version that is functionally identical but using double precision and without additional math functions.

The main kernel that is called from *DetectOutlier* on the host contains an implementation of *EstimateProbability*, along with a number of other functions such as the uniform random number generator, as well as calculation of *pdf* and *cdf*. The current OpenCL framework and the GPU imposes certain additional restrictions, such as a lack of recursion (a function calling itself) and lack of dynamic memory allocation on the GPU. This is a problem as refinement methods used in math libraries often use recursion. These were re-written to remove recursion and to take advantage of additional OpenCL functionality (e.g. the complementary error function *erfc*).

In addition, branching logic can cause a reduction in performance and should be avoided where possible, as GPUs must execute the same code path for each worker thread executing in parallel. As such, counter-intuitive methods such as an arithmetic approach to η-density is used and not removing objects already detected as outliers from further calculation until later are used to avoid branching as far as possible. Memory management is also important on a GPU, with the fastest private memory available for each worker thread used as a scratch area and a slower global memory space that can be accessed by all workers used to store the dataset $\mathcal{D}$. As copying data to and from the GPU is an expensive operation, data transfers are minimized and as much preprocessing and processing done on the GPU as possible.

For optimum performance using a GPU, clearly there must be a high level of data parallelism, with many worker threads to divide the problem. In this implementation, each data object is assigned one worker thread, and each worker is responsible for density sampling of that object's space. To further improve parallelism, vectors are used in each worker's private memory to hold the *pdf* variables. This allows multiple dimensions of each object to be operated on simultaneously on hardware that supports vector operations (4 dimensions in this implementation). The preprocessor will zero additional empty dimensions to ensure the vectors are filled. The simplified overview of the modified *EstimateProbability* (no early reject) that executes on each worker thread is as follows:

EstimateProbabilityWorker(r, s, η, δ)
Let i be the data object of the current worker
Let $F_i^j(\cdot)$ be the inverse cumulative distribution function of *pdf* $h_i^j(\cdot)$
Let vector length $x = r/4$, for vectors of width 4
$prob = 0$
for x times **do**
  Let zero vectors $successes$, $densities$, y be of length x
  $runs = 0$, $subtotal = 0$

```
for s times do
  y = uniformly random variables in [0, 1]
  for all d_k in D do
    densities = densities + h_k^j(F_i^j(y_{1,...,x}))
  end for
  densities = densities/|D|
  successes = successes+ clamp(ceil(densities − η), 0, 1)
  runs = runs + 1
end for
if using 'early reject' then
  for each vector dimension l do
    if (subtotal + successes_l)/l > δ then
      subtotal = (subtotal + successes_l)/l
    else
      successes_l = 0
    end if
  end for
else
  prob = sum(successes)/(r × runs)
end if
end for
Copy prob into global memory for host program
```

Note that while the early reject method adds some overhead to check each dimension against δ, the performance impact is negligible unless the vectors are large relative to the number of objects in the dataset. In our testing, there was no detectable no performance difference between the early reject and no early reject methods.

In addition, the *DetectOutlier* loop that calls *EstimateProbability* is replaced with the following:

```
Copy C_r from the host to the GPU
Call EstimateProbabilityWorker(d, r, s, η, δ) for every object d ∈ C_r
Copy η-probabilities from the GPU back to the host
Add any objects with η-probabilities < δ to O
```

The host loop is thus replaced by multiple workers executing in parallel on the GPU.

The following section shows the results of our testing using a synthetic and a real dataset, as well as the parameters used for optimal results. Note that rather than directly manipulating η and δ, a parameter $uncertainty$ is used both in the generation of synthetic data and to adjust the value of η and δ, compensating for the differences in the underlying standard deviation. When operating on data in which the standard deviation is not known, it can be estimated on a sample of the data by the preprocessor.

5 Experimental Results

The following tests were conducted on a PC running Microsoft Windows Vista SP2 with an Intel Core 2 Duo E8200 dual core CPU and an NVIDIA GeForce GT440 (96

stream processors) GPU. The serial and host code was compiled using Microsoft Visual Studio 2010. The OpenCL code was run using NVIDIA CUDA Toolkit 4.0 and driver 280.26 (OpenCL 1.1) for the GPU, and AMD Stream SDK 2.5 (OpenCL 1.1) for the CPU.

5.1 Performance

To test performance, we generate simple synthetic datasets with a fixed percentage of outliers (10%). In all cases, data objects that are not outliers have a mean value of 0 and a standard deviation of 1, while outliers are offset by 3 (simulating an outlier with high confidence).

In these performance tests, we compare the running time of the serial iterative (CPU-Iterative) and single pass (CPU-Single Pass) methods, as well as the OpenCL implementation on the CPU (CPU-OpenCL) and the GPU. As noted in Sect. 3, the early reject and no early reject methods demonstrated identical performance with the datasets used, so are not shown individually for clarity. All objects in this test have 12 dimensions and use 800 samples per object.

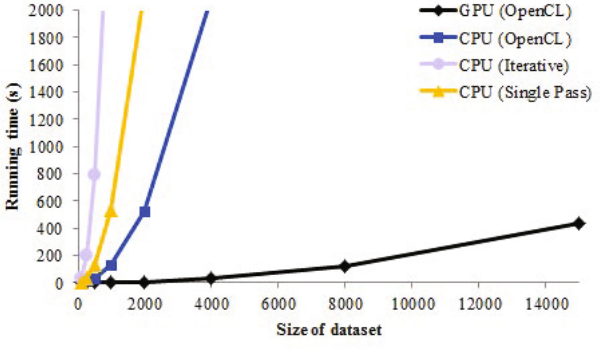


Fig. 1. A comparison of running time with increasing numbers of data objects

It is evident from Fig. 1 that none of the CPU based methods (including the parallel CPU-OpenCL method) offer acceptable performance, with running times increasing rapidly with larger numbers of objects. The GPU offers significant performance improvement over the tested CPU implementations, from 8× at very small sizes against the parallel CPU-OpenCL method up to over 1500× compared against the slowest CPU-Iterative method at larger sizes. Fig. 2 shows the relative increase in running time as dataset size doubles.

The scaling of performance is quadratic with respect to the number of objects in the dataset, due to the algorithm's design. The CPU-based implementations demonstrate this clearly (with some deviation at very small dataset sizes due to overhead).

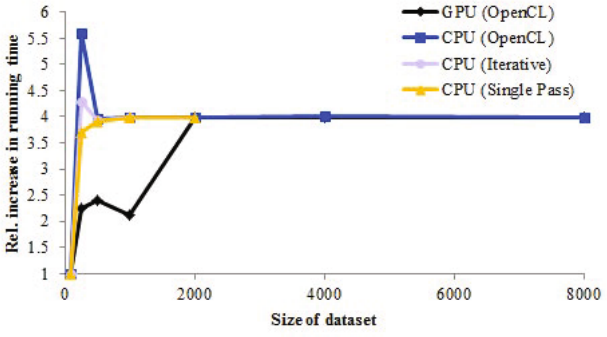


Fig. 2. A comparison of relative increase in running times with each doubling of data objects

At smaller sizes, the GPU demonstrates linear scaling, exceeding the expected quadratic scaling. However the GPU's actual scaling behavior is still quadratic, as the algorithm is unchanged. The processing overhead on the GPU skews performance at smaller dataset sizes, but at sizes exceeding 1000 objects, the worker threads' contention for processing resources and global memory access becomes the performance limiter. It is possible that with the microcluster compression technique described in [8], this behavior can be used advantageously. Overall, the GPU methods maintain a 67× performance improvement over CPU-OpenCL (parallel) and a 273× improvement over CPU-Single Pass (serial).

Figs. 3 and 4 look at two other scaling considerations, the number of dimensions per object and the number of samples per object.

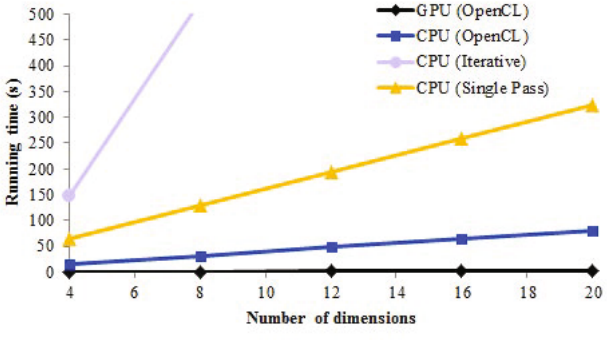


Fig. 3. A comparison of running time with increasing dimensionality

It is clear from Figs. 3 and 4 that the number of dimensions and number of samples offers a strictly linear increase in running time, consistent with the increase in processing load without significant additional memory access load. For the GPU in particular, the increase in running time is negligible.

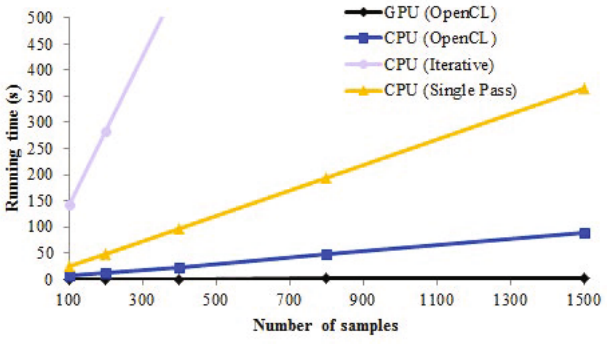


Fig. 4. A comparison of running time with increasing numbers of samples per object

5.2 Quality

Although the focus of this paper has been on performance, the quality of the results must also be acceptable. In the following tests of outlier detection quality, we make the following assumptions: the source data is recorded in a certain form, with data points each having some number of dimensions. To represent the inherent uncertainty, the values of each data point's dimensions are mapped to the mean values of an equivalent uncertain data object's dimensions. The uncertainty of each dimension was estimated from the standard deviation.

To adjust uncertainty in the synthetic dataset, the standard deviation is adjusted in a range from 1 to 3 (i.e. at uncertainty level 3, standard deviation is three times the actual standard deviation). Algorithm parameters η and δ are automatically scaled from 0.3 to 0.6 as uncertainty increases. This is done in an attempt to control the large decline in quality originally seen [8] as uncertainty increases. Although the scaling factors are hard-coded in this implementation, the preprocessor could be extended to better dynamic control of the algorithm parameters and reduce the number of parameters to tune by hand.

The following tests use a real dataset: the Breast Cancer Wisconsin (Diagnostic) Data Set (labeled 'wdbc') from the UCI Machine Learning Repository. This dataset contains 569 records with 30 attributes. This dataset is divided into records marked 'benign' and 'malignant', for the purposes of this test the 'malignant' records are deemed outliers, resulting in a relatively high outlier rate of 37%.

Fig. 5 shows the related parallel and serial methods yielding similar quality, with the CPU methods leading slightly in quality due to the GPU's use of single precision floating point values. Dynamically adjusting the algorithm parameters allows for recall to remain fairly static as uncertainty increases. Note is that the simplest single pass method of averaging density over the entire problem space works well in this relatively small dataset. However as shown in Fig. 6, quality rapidly drops off as more dimensions are added, limiting its usefulness.

For a clearer overview, Fig. 6 represents quality using F1 score, the harmonic mean of precision and recall. It is clear that while adding dimensions results in a slight gain in quality for the iterative and early reject methods, the single pass and no early reject

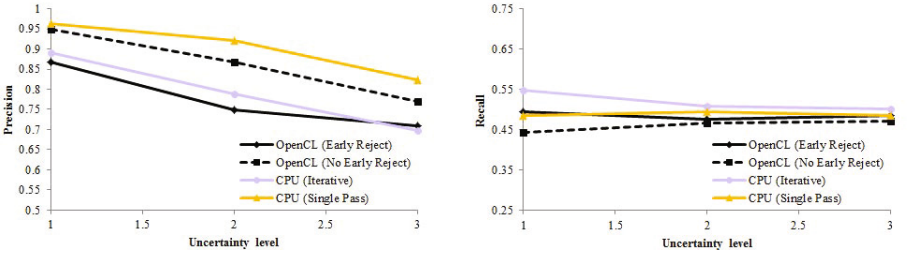


Fig. 5. Precision (left) and Recall (right) with different levels of uncertainty for the 'wdbc' dataset

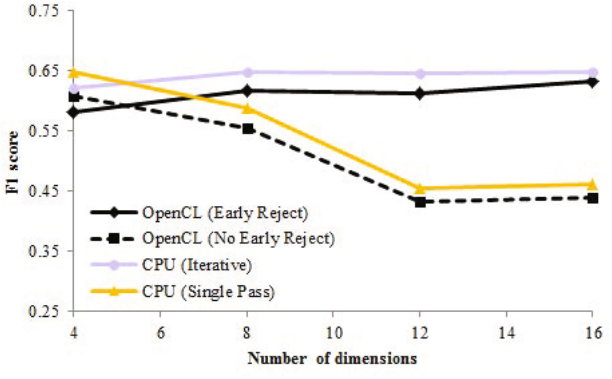


Fig. 6. F1 score with different numbers of dimensions

methods show a significant loss in quality. As outliers are not necessarily of low density in all dimensions, averaging out an object's density over all dimensions leads to areas of low density being lost, thus recall declines significantly.

6 Conclusion

Through this paper, we have demonstrated the use of a density sampling algorithm for outlier detection on uncertain data can be greatly accelerated by leveraging both a GPU and the OpenCL framework. With our implementation, experimental results demonstrate significant reductions to running time from a worst case very small dataset yielding a $8\times$ performance improvement over the parallel CPU-OpenCL implementation and the best case yielding over $1500\times$ improvement compared to the serial CPU-Iterative method. This could enable large numbers of uncertain objects to be scanned in time critical situations, such as in fault detection on sensor networks.

In the future, we would like to explore other techniques to detecting outliers with uncertain data both in comparison to this implementation and in consideration for more methods that can be parallelized for GPU acceleration. Density and distance based calculations are often used in clustering and outlier detection applications, and there are

demonstrable gains from using GPU acceleration in calculation intensive tasks, such as when operating on uncertain data. There are also still opportunities to improve quality and performance (e.g. the microclustering compression technique proposed in [8]), and further testing with more datasets is planned.

Acknowledgments. The work described in this paper was partially supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (PolyU 5191/09E, PolyU A-SA14).

References

1. Aggarwal, C.C. (ed.): Managing and Mining Uncertain Data. Springer (2009)
2. Alshawabkeh, M., Jang, B., Kaeli, D.: Accelerating the local outlier factor algorithm on a gpu for intrusion detection systems. In: Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units (2010)
3. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discoverying clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (1996)
4. Hung, E., Cheung, D.W.: Parallel mining of outliers in large database. Distributed and Parallel Databases 12(1), 5–26 (2002)
5. Tarabalka, Y., Haavardsholm, T.V., Kaasen, I., Skauli, T.: Real-time anomaly detection in hyperspectral images using multivariate normal mixture models and gpu processing. Journal of Real-Time Image Processing 4(3), 287–300 (2009)
6. Bastke, S., Deml, M., Schmidt, S.: Combining statistical network data, probabilistic neural networks and the computational power of gpus for anomaly detection in computer networks. In: 1st Workshop on Intelligent Security (Security and Artificial Intelligence) (2009)
7. Huhle, B., Schairer, T., Jenke, P., Strasser, W.: Robust non-local denoising of colored depth data. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Workshop on Time of Flight Camera based Computer Vision (2008)
8. Aggarwal, C.C., Yu, P.S.: Outlier detection with uncertain data. In: Proceedings of the SIAM International Conference on Data Mining 2008 (2008)
9. Acklam, P.J.: An algorithm for computing the inverse normal cumulative distribution function. Technical report (2003)

Finding Collections of k-Clique Percolated Components in Attributed Graphs

Pierre-Nicolas Mougel[1,2], Christophe Rigotti[1,2], and Olivier Gandrillon[1,3]

[1] Université de Lyon, CNRS, INRIA
[2] INSA-Lyon, LIRIS, UMR5205, F-69621, France
[3] Université Lyon 1, CGPhiMC, UMR5534, F-69622, France

Abstract. In this paper, we consider graphs where a set of Boolean attributes is associated to each vertex, and we are interested in k-clique percolated components (components made of overlapping cliques) in such graphs. We propose the task of finding the collections of homogeneous k-clique percolated components, where homogeneity means sharing a common set of attributes having value true. A sound and complete algorithm based on subgraph enumeration is proposed. We report experiments on two real databases (a social network of scientific collaborations and a network of gene interactions), showing that the extracted patterns capture meaningful structures.

Keywords: graph mining, network analysis, attributed graph, k-clique percolated component.

1 Introduction

During the last decade, graph mining has received an increasing interest in the data mining community. More recently, several works have considered the mining of *enriched* graphs where attributes are associated to the vertices. These works led to interesting results, for instance in clustering [4,8,15,16], dense graph mining [7,12] or graph matching [14].

In this paper, we focus on the special case where the domain of the attributes is Boolean and we propose to extract collections of components called *k-clique percolated components* [1]. More precisely, we define a pattern as a Collection of Homogeneous k-clique Percolated components (CoHoP), where homogeneity means that the vertices in all components share a common set of Boolean attributes having value *true*. A CoHoP pattern must also satisfy two additional constraints: it must contain more than a given number of k-clique percolated components and the vertices must have in common more than a given number of attributes set to *true*. A k-clique percolated component has been defined in [1] as a union of cliques of size k connected by overlaps of $k-1$ vertices (we recall the more formal definition in the next section), and since then, it has been widely accept as one structure that can be used to represent the notion of community. A CoHoP, as introduced here, can thus be interpreted as a set of communities, where elements in all communities share similar Boolean properties.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 181–192, 2012.

In this paper, we also present a sound and complete algorithm to extract the CoHoPs, and we show on two datasets (a coauthor graph and a gene interaction graph) that these patterns can be used to capture useful information, depicting underlying hidden structure of the graph.

The rest of the paper is organized as follows. Section 2 introduces the definition of the CoHoP patterns. The extraction algorithm is described in Section 3 and the experiments are reported in Section 4. The related works are discussed in Section 5, and Section 6 briefly concludes.

2 Pattern Definition

In this section, we first define the dataset structure and recall the notion of k-clique percolated component. Then, we define the targeted patterns, that are collections of k-clique percolated components.

Graphs where information are associated to vertices have been used in different research areas under various names, e.g. attributed graphs [12,14,15,16], itemset-associated graphs [2], informative graphs [4,8], graphs with feature vectors [7]. In this paper, we use the term attributed graphs, and restrict ourselves to Boolean attributed graphs.

Definition 1 (Boolean attributed graph). *A* Boolean attributed graph *is denoted* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{F})$ *where* $\mathcal{V}$ *is the set of vertices,* $\mathcal{E}$ *is the set of edges,* $\mathcal{A}$ *is the set of Boolean attributes, and* $\mathcal{F} : \mathcal{V} \to 2^{\mathcal{A}}$ *is the function returning for a vertex the set of attributes having value true.*

For notational convenience, let us define the following functions.

Definition 2 (Functions *vert* **and** Atb**).** *Let* x *be an attribute. The function* $vert(x) = \{v \in \mathcal{V} \mid x \in \mathcal{F}(v)\}$ *returns the set of vertices having value true for the attribute* x*. Let* M *be a collection of sets of vertices. Then,* $\textsc{Atb}(M) = \bigcap_{V \in M}(\cap_{v \in V}\mathcal{F}(v))$ *is the set of attributes shared by all vertices in* M*.*

Let $\mathcal{G}$ be an attributed graph. We denote $\mathcal{G}[V]$ the subgraph of $\mathcal{G}$ induced by the set of vertices V, i.e., $\mathcal{G}$ restricted to the vertices in V. The notation $\mathcal{G}[\![X]\!]$ denotes the subgraph induced by the set of vertices having value true for all attributes in X, i.e., $\mathcal{G}[\![X]\!] = \mathcal{G}[\cap_{x \in X} vert(x)]$.

A clique is a set of vertices in which every pair of distinct vertices is connected by an edge and a k-clique is a clique of size k. A k-clique percolated component (termed also k-clique-community in [11]) is a relaxed version of the concept of cliques. The definition of k-clique percolated component given in [1] can be reformulated as follows using an equivalence relation over the cliques.

Definition 3 (Adjacency relation). *Let* $\mathcal{G}$ *be an attributed graph and* $\mathfrak{R}$ *be the adjacency relation over the* k*-cliques in* $\mathcal{G}$*. Two* k*-cliques are related by* $\mathfrak{R}$ *if and only if they have an intersection of at least* $k-1$ *vertices. Let* $\mathfrak{R}^t$ *be the transitive closure of* $\mathfrak{R}$*.*

The relation $\mathfrak{R}$ is symmetric and reflexive, so $\mathfrak{R}^t$ is symmetric, reflexive, and transitive. Consequently, $\mathfrak{R}^t$ is an equivalence relation.

Definition 4 (k-clique Percolated Component (k-PC)). *A k-PC is the union of all k-cliques in a class of equivalence over $\mathfrak{R}^t$.*

In other words, a k-PC is the union of all k-cliques that can be reached from each other through a series of adjacent k-cliques. We will denote $\mathcal{C}_{kpc}(\mathcal{G})$ the collection of all k-PCs in an attributed graph $\mathcal{G}$. Compared to other fault-tolerant clique definitions, the particularity of k-PC is to enforce the fact that each vertex can be reached from any other vertex through well connected subset of vertices [11]. In the context of social networks, it represents a community where each person, even if not directly connected to another member, can easily find a way to communicate with him/her. Note also that with this definition, a clique is contained in at most one k-PC. However, since a vertex can be in several cliques sharing less than $k-1$ vertices, it can be part of several k-PCs.

As mentioned in the introduction, our purpose is to explore the relation between strongly connected subgraphs. To perform this task we extract collections of set of vertices such that, with k, α, and γ three user defined positive integers, (1) all vertices are homogeneous, more precisely, they have at least α true-valued attributes in common, (2) the collection contains at least γ k-PCs and (3) all k-PCs showing the same true-valued attributes are in the collection. These patterns are defined more precisely as follows.

Definition 5 (Collection of Homogeneous k-PCs (CoHoP)). *Let k, α, and γ be three strictly positive integers, and $\mathcal{G}$ be an attributed graph. A collection M of sets of vertices is a CoHoP if and only if:*

- $|\textsc{Atb}(M)| \geq \alpha$ *(the vertices in M are homogeneous);*
- *M contains at least γ k-PCs, i.e., $|M| \geq \gamma$;*
- *M is the collection of all k-PCs in $\mathcal{G}[\![\textsc{Atb}(M)]\!]$, i.e., M contains* all *k-PCs sharing the attributes in* Atb*(M).*

Note that due to the constraint on homogeneity, a k-PC which is formed by vertices sharing less than α attributes cannot be part of a CoHoP.

3 Mining CoHoP Patterns

We first present a naive algorithm enumerating all subgraphs which might contain a pattern. Then we show how to safely reduce the subgraphs enumeration, and we describe the corresponding algorithm. Finally we describe implementation techniques.

Naive Algorithm

While Definition 5 is very declarative, we establish the following more constructive definition of the CoHoPs.

Lemma 1. *Let k, α, and γ be three strictly positive integers, and $\mathcal{G}$ be an attributed graph with $\mathcal{A}$ the set of Boolean attributes in $\mathcal{G}$. A collection M of sets of vertices is a CoHoP if and only if there exists $X \subseteq \mathcal{A}$ such that $M = \mathcal{C}_{kpc}(\mathcal{G}[\![X]\!])$, $|X| \geq \alpha$, and $|M| \geq \gamma$.*

Proof. First, consider a CoHoP M. By direct application of Definition 5, there exists $X = \text{ATB}(M) \subseteq \mathcal{A}$ such that $M = \mathcal{C}_{kpc}(\mathcal{G}[\![X]\!])$, $|X| \geq \alpha$, and $|M| \geq \gamma$. Now we prove the reciprocal. Consider X a set of attributes satisfying $|X| \geq \alpha$, and M a collection of sets of vertices such that $M = \mathcal{C}_{kpc}(\mathcal{G}[\![X]\!])$ and $|M| \geq \gamma$. Since $M = \mathcal{C}_{kpc}(\mathcal{G}[\![X]\!])$, then $X \subseteq \text{ATB}(M)$. So $\mathcal{G}[\![\text{ATB}(M)]\!]$ is a subgraph of $\mathcal{G}[\![X]\!]$ and since all vertices in M are also in $\mathcal{G}[\![\text{ATB}(M)]\!]$, we have $M = \mathcal{C}_{kpc}(\mathcal{G}[\![X]\!]) = \mathcal{C}_{kpc}(\mathcal{G}[\![\text{ATB}(M)]\!])$. Thus M is a CoHoP.

To compute all patterns, a naive algorithm can enumerate the subgraphs $\mathcal{G}_e = \mathcal{G}[\![X]\!]$ for all non empty set of attributes X, and for each $\mathcal{G}_e$ computes all k-PCs in $\mathcal{G}_e$. Then, if $|X| \geq \alpha$ and if there is at least γ k-PCs in $\mathcal{G}_e$, this collection of k-PCs is a CoHoP. From Lemma 1, this algorithm is correct. However, with this enumeration technique, $2^{|\mathcal{A}|} - 1$ subgraphs will have to be enumerated ($2^{|\mathcal{A}|} - 1$ non empty subsets of $\mathcal{A}$). The following lemmas are used to reduce the collection of subgraphs that has to be enumerated.

Reducing the Collection of Graphs to Be Enumerated

First, we introduce the notion of k-max-clique which is a clique having at least k vertices and not being a subset of any other clique. The collection of all k-max-cliques in an attributed graph $\mathcal{G}$ is denoted $\mathcal{C}_{kmax}(\mathcal{G})$.

The next lemma states that we can discard the attributed graphs that do not contain at least γ k-max-cliques, and also their subgraphs.

Lemma 2. *Let $\mathcal{G}$ be an attributed graph. If $\mathcal{G}$ does not contain at least γ k-max-cliques, then neither $\mathcal{G}$ nor any subgraph of $\mathcal{G}$ can contain a CoHoP.*

Proof. Let $\mathcal{G}$ be an attributed graph having less than γ k-max-cliques. Since all k-cliques in a k-max-clique are in the same k-PC, then the number of k-max-cliques cannot be greater than the number of k-PCs. So, $\mathcal{G}$ cannot contain γ k-PCs and thus cannot contain a CoHoP. The same holds for any subgraph of $\mathcal{G}$, since a subgraph of $\mathcal{G}$ cannot contain more k-max-cliques than $\mathcal{G}$.

According to the following lemma, we can avoid the enumeration of graphs (and their subgraphs) if they are induced by sets of attributes shared by not enough vertices to contain a CoHoP.

Lemma 3. *Let $\mathcal{G}$ be an attributed graph and x an attribute shared by less than k vertices in $\mathcal{G}$. Then, the graph $\mathcal{G}[\![\{x\}]\!]$ and all its subgraphs cannot contain a CoHoP.*

Proof. Straightforward since $\mathcal{G}[\![\{x\}]\!]$ contains less than k vertices.

The following property allows to reduce the set of vertices under consideration.

Lemma 4. *Let $\mathcal{G}$ be an attributed graph. Only vertices in a k-max-clique of $\mathcal{G}$ can form a CoHoP in $\mathcal{G}$ or in any subgraph of $\mathcal{G}$.*

Proof. Direct, as a vertex which is not in a k-max-clique cannot be in any k-PC.

Algorithm Description

A recursive function `FindCoHoP`, that takes advantage of Lemmas 2, 3, and 4 to prune the search space, is presented as Algorithm 1. The input of the algorithm for the first call is the whole attributed graph, i.e., $\mathcal{G}_e = \mathcal{G}$, and $\mathcal{A}_c$, the set of candidate attributes remaining under consideration to find attributes shared by subgraph, is $\mathcal{A}$.

Line 1 checks that there is at least γ k-max-cliques in $\mathcal{G}_e$. If it is not the case, from Lemma 2 no subgraph of $\mathcal{G}_e$ including $\mathcal{G}_e$ itself can contain a k-PC. **Line 2** computes the set of vertices which might contain a k-PC (i.e., $\mathcal{V}_r$) as the union of all k-max-cliques in $\mathcal{G}_e$ according to Lemma 4. **Line 3** checks (1) if there is at least α attributes shared by all vertices in $\mathcal{V}_r$ ($|\cap_{v \in \mathcal{V}_r} \mathcal{F}(v)| \geq \alpha$) and (2) if there is at least γ k-PCs ($|\mathcal{C}_{kpc}(\mathcal{G}_e[\mathcal{V}_r])| \geq \gamma$). If so, the collection of k-PCs is a CoHoP, and is output on **line 4**. On **line 5**, attributes from $\mathcal{A}_c$ shared by all vertices in $\mathcal{V}_r$ are removed from $\mathcal{A}_c$. Removing these attributes does not change the collection of enumerated subgraphs, since if we pick such an attribute x we have $\mathcal{G}_e[\mathcal{V}_r \cap vert(x)]$ that is equal to $\mathcal{G}_e[\mathcal{V}_r]$ itself in the recursive call to `FindCoHoP` (**line 9**). On **line 6**, attributes shared by less than k vertices in $\mathcal{V}_r$ are removed from $\mathcal{A}_c$, according to Lemma 3. This avoids unnecessary calls to `FindCoHoP` with subgraphs having not enough vertices. **Lines 7 to 9** perform a standard recursive enumeration scheme to produce in a depth first way, and element by element (the x that is picked), all subsets of $\mathcal{A}_c$. While $\mathcal{A}_c$ is not empty, an attribute x is picked (**line 8**) and function `FindCoHoP` is called with the subgraph of $\mathcal{G}_e$ induced by the set of vertices in $\mathcal{V}_r$ sharing attribute x, i.e., $\mathcal{G}_e[\mathcal{V}_r \cap vert(x)]$.

Algorithm 1. `FindCoHoP`

Input: $\mathcal{G}_e$, $\mathcal{A}_c$
1 **if** $|\mathcal{C}_{kmax}(\mathcal{G}_e)| \geq \gamma$ **then**
2 $\mathcal{V}_r = \cup_{C \in \mathcal{C}_{kmax}(\mathcal{G}_e)}$
3 **if** $|\cap_{v \in \mathcal{V}_r} \mathcal{F}(v)| \geq \alpha$ **and** $|\mathcal{C}_{kpc}(\mathcal{G}_e[\mathcal{V}_r])| \geq \gamma$ **then**
4 **output** $\mathcal{C}_{kpc}(\mathcal{G}_e[\mathcal{V}_r])$
5 $\mathcal{A}_c \leftarrow \{x \in \mathcal{A}_c \mid \mathcal{V}_r \nsubseteq vert(x)\}$
6 $\mathcal{A}_c \leftarrow \{x \in \mathcal{A}_c \mid |vert(x) \cap \mathcal{V}_r| \geq k\}$
7 **while** $\mathcal{A}_c \neq \emptyset$ **do**
8 Pick and remove an attribute x from $\mathcal{A}_c$
9 `FindCoHoP`$(\mathcal{G}_e[\mathcal{V}_r \cap vert(x)], \mathcal{A}_c)$

Theorem 1. *Algorithm 1 returns all CoHoP patterns and only CoHoP patterns.*

Proof. Lemma 1 and Lemmas 2 to 4 (safety of the pruning) ensure the completeness of Algorithm 1. Line 3 ensures its soundness.

Note that a given CoHoP might be output several times by Algorithm 1. Such duplicates are removed in a simple post-processing step.

Implementation
We give here some details about the implementation of Algorithm 1 used in the experiments presented in the next section.

The algorithm used to compute the collection of k-PCs in a graph is the one described in [11] and also used for instance in [3]. It first builds a matrix representing the adjacency relation between the k-cliques, and then compute the connected components of k-cliques (the k-PCs) using this matrix. The algorithm used to compute the k-max-cliques is CLIQUES [13]. Both the collection of k-max-cliques (i.e., $\mathcal{C}_{kmax}(\mathcal{G}_e)$) and the collection of k-PCs (i.e., $\mathcal{C}_{kpc}(\mathcal{G}_e[\mathcal{V}_r])$) are computed only once for a given attributed graph on respectively lines 1 and 3, and are reused on lines 2 and 4. Moreover, the computation of the k-PCs is done on line 3 only if the vertices in $\mathcal{V}_r$ have at least α attributes in common (i.e., $|\cap_{v\in\mathcal{V}_r}\mathcal{F}(v)| \geq \alpha$).

Finally, since vertices in a pattern must share at least one attribute ($\alpha \geq 1$), in general it is not necessary to compute the k-max-cliques of the whole graph. So, the first level of the enumeration is computed using only lines 6 to 9 of Algorithm 1, with $\mathcal{V}_r$ the set of all vertices of the input attributed graph.

4 Experiments

In this section we report experiments on three datasets built using real data: two bibliographic datasets (DBLP_1 and DBLP_2), and a biological dataset (BioData). The size and density of these datasets are presented in Table 1. All experiments were performed on a PC running GNU/Linux with a 3 GHz Core 2 Duo CPU and 8 GB of main memory installed (no more than 2 GB where used). We first describe the datasets, then, we illustrate the interest of the CoHoPs by mean of three typical examples of pattern found. Next, we discuss the performances of the algorithm and parameter setting.

Collaboration Network: DBLP_1 and DBLP_2 datasets have been built using the public DBLP database[1]. This database contains rather exhaustive bibliographic information on most computer science conferences and journals. We built our datasets using all conferences and journal up to august 2011. A vertex corresponds to an author and the attributes associated to a vertex are the conferences and journals in which the author has published. An edge between two authors represents the fact that they have coauthored some papers.

For DBLP_1 we wanted a large dataset to assess the performances of extraction algorithm. Consequently, in DBLP_1, we put an edge to represent each pair of coauthors, and for the attributes, an author is associated to all conferences and journals in which she/he has published. This led to a dataset containing $997,050$ authors and $5,963$ conferences/journals.

The dataset DBLP_2 was targeted to obtained more meaningful patterns. Thus, in DBLP_2, we focused on pairs of authors that have been collaborating more significantly, and we put an edge between two authors if they were coauthors of at least three articles. We also required a stronger relationship between authors

[1] http://dblp.uni-trier.de/

and conferences/journals. Indeed, we associated a conference or a journal to an author, only if this author has published at least three times in this conference/journal. Finally, in $DBLP_2$, authors that remain associated to no conference/journal (i.e., authors who have never published three times in the same conference/journal) were removed.

Protein Interaction Network: BioData has been built using two databases STRING[2] [5] and SQUAT[3] [6]. STRING integrates data on protein-protein interactions from different sources (e.g., genomic data, co-expression, experiments, literature). Among these interactions we only retained interactions with *confidence*[4] higher or equal to 400 (default STRING selection threshold). SQUAT is a public database of Boolean gene expression data resulting from SAGE experiments. SQUAT contains for thousands of genes, the sets of biological situations (termed *libraries*) where these genes are overexpressed. In our experiments, only *Human species* genes were used. We built the BioData dataset as follows. A vertex is a gene, and we put an edge between two genes if there was an interaction reported in STRING (confidence of at least 400) between the proteins corresponding to these two genes. The attributes associated to a gene were simply the biological situations in which the gene was overexpressed according to the SQUAT database. In our experiments, only Human species genes were used, this led to $15,571$ genes common to the two databases. For these genes we have expression data in SQUAT for 486 different biological situations.

Table 1. Size and density of datasets $DBLP_1$, $DBLP_2$, and BioData

	$DBLP_1$	$DBLP_2$	BioData
# Vertices	997,050	127,386	15,571
# Attributes	5,963	3,980	486
Avg. degree	6.88	3.69	20.01
Avg. attributes/vertex	3.06	2.15	11.46

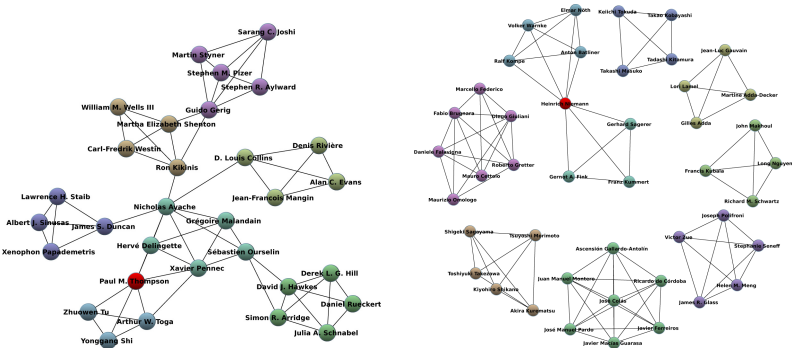
4.1 Illustration of the Interest of the Patterns

Collaboration Network: Let us first define some vocabulary in the context of a network of researchers. In [11] the authors consider that a k-PC is a community in the sense that "it consists of several complete subgraphs that tend to share many of their nodes". Consequently, we will use the term community for a k-PC. We will also say that two communities are connected if there is an edge between both communities. In $DBLP_2$, we searched for CoHoPs with at least seven 4-PCs were all authors have published in the same three conferences or journals

[2] http://string-db.org/

[3] http://bsmc.insa-lyon.fr/squat/

[4] This confidence is a measure provided by STRING. Low confidence means that there are not so many evidences that the interaction exists.



(a) Seven 4-PCs concerning conferences IPMI, ISBI, MICCAI and journal IEEE Trans. Med. Imaging.

(b) Nine 4-PCs concerning conferences INTERSPEECH, ICSLP, and EUROSPEECH.

Fig. 1. Two patterns extracted from DBLP_2 with $k = 4$, $\gamma = 7$, and $\alpha = 3$. Each color corresponds to a k-PC. Vertices in red are in several k-PCs

(i.e., $k = 4$, $\alpha = 3$ and $\gamma = 7$). With this parameter setting, 57 CoHoPs where extracted. To illustrate the kind of patterns that were retrieved, we focus on two patterns presented in Figures 1(a) and 1(b).

The pattern on Figure 1(a) contains seven 4-PCs, all authors having published in conferences or journals related to medical imaging. The authors N. Avache, H. Delingette, G. Malandain, S. Ourselin, X. Pennec, and P. M. Thompson are forming a community connected to all other communities except one and is the core of a star-based topology. Knowing such a structure is useful to make some decisions. For instance having researchers of the core community as partners in a project, or choosing this community as a destination for a post-doc position could be a great opportunity to benefit from contacts with all the other groups. We also investigated the role of the authors connecting two communities (i.e., the endpoints of edges connecting two communities) in this pattern using ArnetMiner[5]. We found that four of these *bridging nodes* [10] were advisor of at least half of the authors of their respective communities. So they are likely to be senior researchers and this is coherent with the fact that they appear as bridges between communities.

In the second CoHoP, presented on Figure 1(b), all authors have published at least three times in three conferences related to speech communication / spoken language. It contains nine communities, seven of them not being connected to any other. Moreover, from the personal page of the authors, we found out that in most cases a community is formed by people working in the same research institute. So, here most communities are formed by researchers working in the field of speech processing and not strongly publishing with researchers from other institutes. Such structure with disconnected groups of people sharing

[5] ArnetMiner (`http://arnetminer.org/`) is an application providing the relationship (e.g., coauthor, advisor, advisee) between researchers.

similar interests might be interesting for several tasks. For instance, it can give hints to funding organisms to set up long term development strategies of collaboration networks. Or it can also be helpful, in a normal day-to-day activity, like finding reviewers for a paper, by suggesting experts in the same domain as the authors, but having no closed collaborations (strong coauthor relationship) with these authors (and also eventually having no closed collaborations with the other experts).

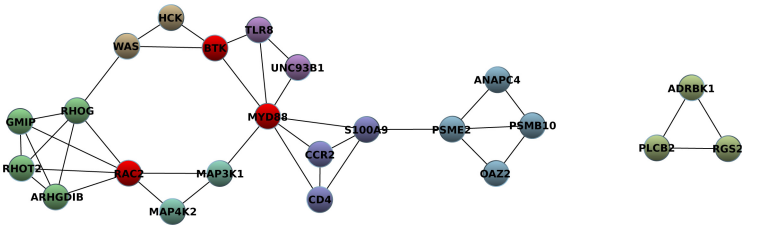


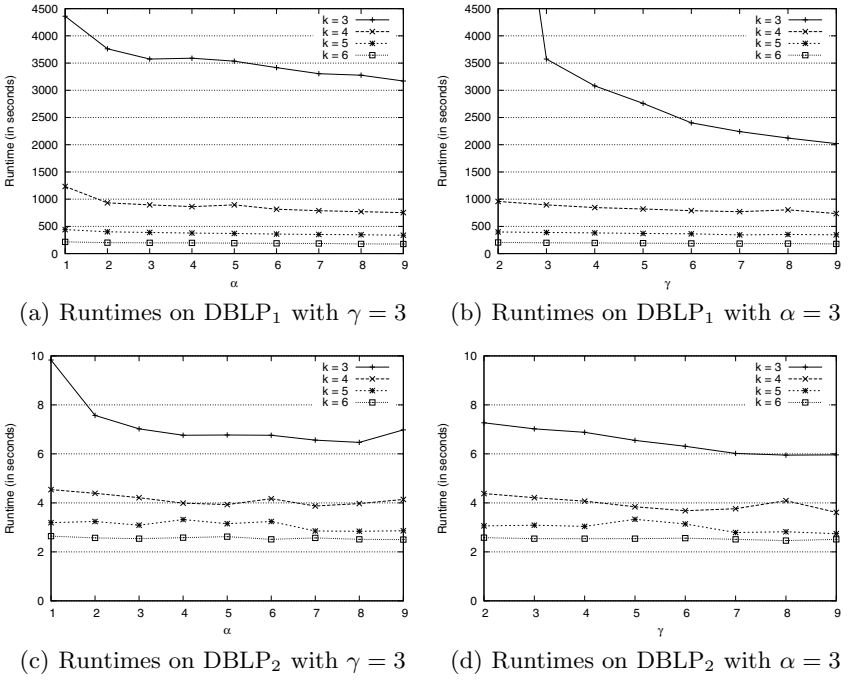
Fig. 2. A CoHoP extracted from BioData with $k = 3$, $\alpha = 4$, and $\gamma = 3$. Each color corresponds to a k-PC. Vertices in red are in several k-PCs.

Protein Interaction Network: In the BioData dataset, we searched for CoHoPs with at least three 3-PCs were all genes are overexpressed in at least four biological situations (i.e., $k = 3$, $\alpha = 4$, and $\gamma = 3$) and obtained 25 patterns. The CoHoP containing the greatest number of k-PCs is presented Figure 2. This CoHoP is composed of 7 k-PCs, and all vertices are genes overexpressed in 4 situations corresponding to normal white blood cells activities. The pattern contains (from left to right) a ring made of 4 groups of genes (4 k-PCs), with two other groups forming a tail link to the ring, and an extra isolated group. Such a structure suggest, among others, the following biological questions. Is there any order in the activation of the groups along the ring ? Do the groups forming the tail act as a trigger for the whole ring activity ? Are there some interactions between the isolated group and the others (while no such interaction is reported in STRING with a confidence of 400 or greater) ? All these questions can lead to interesting deeper investigations through wet biology experiments.

4.2 Performance Study

Figure 3 shows that the extraction can be made in less than 25 minutes when $k \geq 4$ on DBLP_1 and DBLP_2 (on BioData, all extractions made for similar settings were run in less than 10 seconds). We can also notice that in all settings, the runtime increases significantly when k decreases. The runtime increase when γ decreases, as shown on Figure 3(b), is mainly due to the computation of k-PCs from a large number of k-max-cliques.

The number of output patterns is given on Figure 4. As expected this number decreases when the values of k, α, and γ increase. Such curves can be used to help setting the extraction parameters. For instance, for communities, the literature [11,3] recommends to use a value of k between 3 and 6. So, for DBLP_2,



(a) Runtimes on $DBLP_1$ with $\gamma = 3$ (b) Runtimes on $DBLP_1$ with $\alpha = 3$

(c) Runtimes on $DBLP_2$ with $\gamma = 3$ (d) Runtimes on $DBLP_2$ with $\alpha = 3$

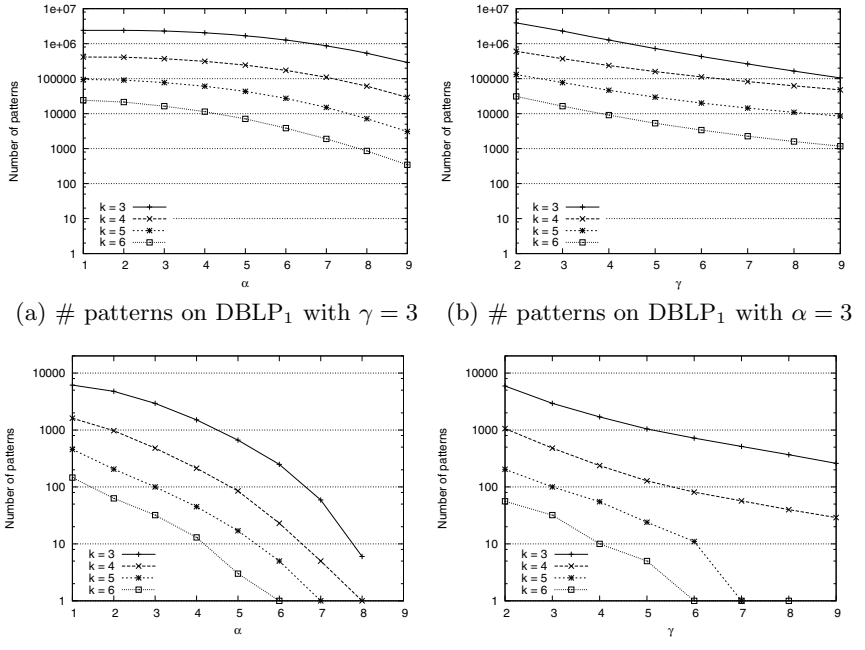
Fig. 3. Runtime for different sets of parameters on $DBLP_1$ and $DBLP_2$

since the running time is rather low, we could count the number of patterns for these values of k and a whole range of values of α and γ. Then we chose among these settings the ones that were meaningful and that lead to collections of patterns of reasonable size (for human browsing).

5 Related Work

Local pattern mining in attributed graphs to find homogeneous set of vertices is rather recent, and two main families of approaches have been developed.

In the first family [7,12], a pattern is a *single* densely connected subgraph (e.g., a quasi-clique) such that the vertices have homogeneous feature values. Such a pattern can reveal a module, a group or a community sharing similar properties or interests. A pattern in our approach is *set* of groups sharing similar attribute values, and thus exhibits a different kind of structures made of several groups (not a single one). Moreover the notion of group is also different, and corresponds for CoHoPs to an another well known form of communities, the k-clique percolated components. It should also be pointed out, that if the user is interested in extracting single groups, this can also be done, in the case of CoHoPs, by setting parameter γ to 1 and by outputting all k-clique percolated components as separated patterns.



(a) # patterns on DBLP_1 with $\gamma = 3$ (b) # patterns on DBLP_1 with $\alpha = 3$

(c) # patterns on DBLP_2 with $\gamma = 3$ (d) # patterns on DBLP_2 with $\alpha = 3$

Fig. 4. Number of patterns for different sets of parameters on DBLP_1 and DBLP_2

Our proposal is closer to the second family of approaches [2,9], where a pattern is a *collection* of set of vertices in a subgraph made of vertices sharing similar attribute values. These previous works adopt opposite views on the kind of structures they consider. In [9] the constraint on the structure of a group is very strong, since the sets of vertices must be cliques. On the contrary, in [2], the choice was made to be very tolerant, since a set of vertices is simply required to form a connected subgraph. We introduce in this paper a complementary approach, that exhibits another kind of group structures, namely the k-clique percolated components, that are typical group structures used in the literature to capture the notion of community.

6 Conclusion

In this paper, we considered graphs having a set of Boolean attributes associated to each vertex. We proposed to find Collection of Homogeneous k-clique Percolated components (CoHoP) and gave a sound and complete algorithm for this task. We shown by means of experiments on real datasets that the extractions can be made in practice and lead to meaningful patterns.

Acknowledgments. This work is partly funded by the Rhône-Alpes Complex Systems Institute (IXXI) through the REHMI project, and by the French National Research Agency (ANR) through the projects FOSTER (ANR-2010-COSI-012-02).

References

1. Derényi, I., Palla, G., Vicsek, T.: Clique percolation in random networks. Phys. Rev. Lett. 94, 160–202 (2005)
2. Fukuzaki, M., Seki, M., Kashima, H., Sese, J.: Finding Itemset-Sharing Patterns in a Large Itemset-Associated Graph. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010, Part II. LNCS, vol. 6119, pp. 147–159. Springer, Heidelberg (2010)
3. Gao, W., Wong, K.-F., Xia, Y., Xu, R.: Clique Percolation Method for Finding Naturally Cohesive and Overlapping Document Clusters. In: Matsumoto, Y., Sproat, R.W., Wong, K.-F., Zhang, M. (eds.) ICCPOL 2006. LNCS (LNAI), vol. 4285, pp. 97–108. Springer, Heidelberg (2006)
4. Ge, R., Ester, M., Gao, B.J., Hu, Z., Bhattacharya, B., Ben-Moshe, B.: Joint cluster analysis of attribute data and relationship data: The connected k-center problem. ACM Trans. Knowl. Discov. Data (TKDD) 2(2), 1–35 (2008)
5. Jensen, L.J., Kuhn, M., Stark, M., Chaffron, S., Creevey, C., Muller, J., Doerks, T., Julien, P., Roth, A., Simonovic, M., Bork, P., von Mering, C.: STRING 8-a global view on proteins and their functional interactions in 630 organisms. Nucleic Acids Research 37, 412–416 (2009)
6. Leyritz, J., Schicklin, S., Blachon, S., Keime, C., Robardet, C., Boulicaut, J.F., Besson, J., Pensa, R.G., Gandrillon, O.: Squat: A web tool to mine human, murine and avian sage data. BMC Bioinformatics 9(1), 378 (2008)
7. Moser, F., Colak, R., Rafiey, A., Ester, M.: Mining cohesive patterns from graphs with feature vectors. In: SIAM Data Mining Conf (SDM), pp. 593–604 (2009)
8. Moser, F., Ge, R., Ester, M.: Joint cluster analysis of attribute and relationship data without a-priori specification of the number of clusters. In: Int. Conf. on Knowledge Discovery and Data Mining (KDD), p. 510 (2007)
9. Mougel, P.N., Plantevit, M., Rigotti, C., Gandrillon, O., Boulicaut, J.F.: Constraint-Based Mining of Sets of Cliques Sharing Vertex Properties. In: Workshop on Analysis of Complex NEtworks (ACNE 2010) co-located with ECML/PKDD 2010 (2010)
10. Musiał, K., Juszczyszyn, K.: Properties of Bridge Nodes in Social Networks. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 357–364. Springer, Heidelberg (2009)
11. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435(7043), 814–818 (2005)
12. Silva, A., Meira Jr., W., Zaki, M.J.: Structural correlation pattern mining for large graphs. In: Workshop on Mining and Learning with Graphs (MLG), pp. 119–126 (2010)
13. Tomita, E., Tanaka, A., Takahashi, H.: The worst-case time complexity for generating all maximal cliques and computational experiments. Theor. Comput. Sci. (TCS) 363, 28–42 (2006)
14. Tong, H., Gallagher, B., Faloutsos, C., Eliassi-rad, T.: Fast best-effort pattern matching in large attributed graphs. In: Int. Conf. on Knowledge Discovery and Data Mining, KDD (2007)
15. Zhou, Y., Cheng, H., Yu, J.X.: Graph clustering based on structural/attribute similarities. Proc. VLDB Endow. 2, 718–729 (2009)
16. Zhou, Y., Cheng, H., Yu, J.X.: Clustering large attributed graphs: An efficient incremental approach. In: Int. Conf. on Data Mining (ICDM), pp. 689–698 (2010)

Reciprocal and Heterogeneous Link Prediction in Social Networks

Xiongcai Cai, Michael Bain, Alfred Krzywicki, Wayne Wobcke, Yang Sok Kim, Paul Compton, and Ashesh Mahidadia

School of Computer Science and Engineering,
The University of New South Wales, Sydney, NSW 2052, Australia
{xcai,mike,alfredk,wobcke,yskim,compton,ashesh}@cse.unsw.edu.au

Abstract. Link prediction is a key technique in many applications in social networks, where potential links between entities need to be predicted. Conventional link prediction techniques deal with either homogeneous entities, e.g., people to people, item to item links, or non-reciprocal relationships, e.g., people to item links. However, a challenging problem in link prediction is that of heterogeneous and reciprocal link prediction, such as accurate prediction of matches on an online dating site, jobs or workers on employment websites, where the links are reciprocally determined by both entities that heterogeneously belong to disjoint groups. The nature and causes of interactions in these domains makes heterogeneous and reciprocal link prediction significantly different from the conventional version of the problem. In this work, we address these issues by proposing a novel learnable framework called *ReHeLP*, which learns heterogeneous and reciprocal knowledge from collaborative information and demonstrate its impact on link prediction. Evaluation on a large commercial online dating dataset shows the success of the proposed method and its promise for link prediction.

Keywords: Machine Learning, Data Mining, Information Retrieval, Recommender Systems.

1 Introduction

Social networks are commonly used to model the interactions among people in communities, which can be represented by graphs where a vertex corresponds to a person in some community and an edge or *link* represents some association between the corresponding people. Understanding the association between two specific vertices by predicting the likelihood of a future but not currently existing association between them is a fundamental problem known as *link prediction* [13].

Social interaction on the Web often involves both positive and negative relationships, e.g., since attempts to establish a relationship may fail due to rejection from the intended target. This generates links that signify rejection of invitations, disapproval of applications, or expression of disagreement with others' opinions. Such social networks are *reciprocal* since the sign of a link indicating

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 193–204, 2012.

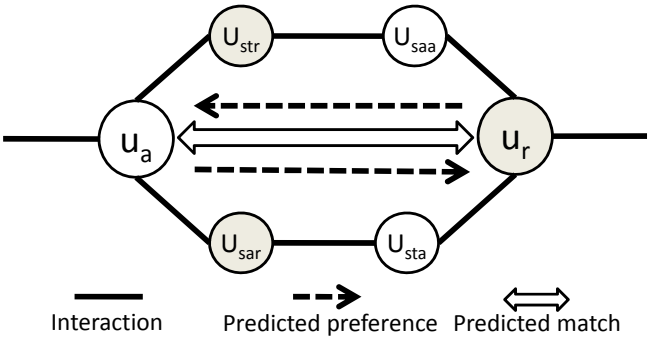


Fig. 1. Collaborative Information for Reciprocal and Heterogeneous Link Prediction. Links as interactions connect nodes (in circles) that belong to different groups (grey shading or no shading). The signs of links represent matches of two way decisions, which are predicted by a learning model using collaborative information.

whether it is positive or negative depends on the attitudes or opinions of both entities forming the link. While the interplay of positive and negative relations is clearly important in many social network settings, the vast majority of online social network research has considered only positive relationships. Moreover, *reciprocal* positive and negative relationships have been even less investigated. Recently, social network analysis has had a variety of applications, such as online dating sites, education admission portals as well as jobs, employment, career and recruitment sites, where people in the networks have different roles, and links between them can only be between people in different roles. Such networks are *heterogeneous*, creating challenges for link prediction since existing approaches focus only on homogeneous networks where nodes in the networks have the same role and any of them may link to any other.

In this work, we consider the heterogeneous and reciprocal link prediction problem. We propose a framework to address prediction of the sign of a link in heterogeneous and reciprocal networks. We model this problem as a machine learning problem and create structural features for learning, i.e. we construct features for learning based on structural collaborative information. Specifically, motivated by *taste* and *attractiveness* in the Social Collaborative Filter [4], we first define a structural unit called a *tetrad* (to be defined in Section 4.1), i.e. a path crossing four nodes as in Figure 1 [4], in the graph of networks based on a set of variations of collaborative filtering. These represent collaborative information regarding taste and attractiveness of nodes in the graph (people in social networks). The properties of each tetrad are then measured in terms of positive and negative signs through its path. Finally, the properties of a tetrad are used as features in a learning framework for link sign prediction.

The paper is organised as follows. Section 2 presents related work. Section 3 defines the problem. Section 4 develops a learnable framework for the reciprocal

and heterogeneous link prediction problem. Experimental evaluation is in Section 5 and we conclude in Section 6.

2 Related Work

Liben-Nowell and Kleinberg [13] developed one of the earliest link prediction models for social networks. They concentrated mostly on the performance of various graph-based similarity metrics for the link prediction problem. This work has since been extended to use supervised learning for link prediction [8,2,7], where link prediction was considered as a binary classification task in a supervised learning setup using features extracted from various network properties.

Recent developments in online social networks such as Facebook and Twitter have raised scalability challenges for link prediction. Large scale link prediction was addressed by Acar et al. in [1], where higher-order tensor models based on matrix factorisation were used for link prediction in large social networks.

Recently, work on link prediction has started considering both negative and positive relationships in online websites [12,3,10,11]. Leskovec et al. investigated the problem of link sign prediction to uncover the mechanisms that determine the signs of links in large social networks where interactions can be both positive and negative [12]. Also, learning methods based on multiple sources and multiple path-based features were investigated. In [6], there is a collective link prediction problem where several related link prediction tasks are jointly learned. In [14], a supervised learning framework has been designed based on a rich variety of path-based features using multiple sources to learn the dynamics of social networks.

In reciprocal and heterogeneous link prediction, the reciprocal and heterogeneous nature of networks makes the problem significantly different from traditional link prediction. Therefore, new methods to: 1) model the characteristics of how reciprocal and heterogeneous links form; and 2) that can be used for mining and predicting such links in large social network datasets are essential.

3 Problem Statement

Link prediction is defined as the inference of new interactions among the members of a given social network [13]. More formally, the link prediction problem is defined as: given a snapshot of a social network at time t, seek to accurately predict the edges that will be added to the network during the interval from time t to a given future time $t' = t + \delta t$. The solution to this problem lies in modelling the evolution of the network using intrinsic features derived from the network itself in order to understand which properties of the network lead to the most accurate link predictions.

Edge sign prediction is an important type of link prediction, defined as follows. Suppose we are given a social network with signs on all its edges, but the sign on the edge from node u to node v, denoted $s(u, v)$, has been hidden. How reliably can we infer this sign $s(u, v)$ using the information provided by the rest of the network? This problem is both a concrete formulation of our basic questions

about the typical patterns of link signs, and also a way of approaching our motivating application of inferring unobserved attitudes among users of social computing sites. For a given link in a social network, we will define its sign to be positive or negative depending on whether it expresses a positive or negative attitude from the source of the directed link to the target, and *vice versa*.

Heterogeneous and reciprocal link prediction deals with predictions for heterogeneous and reciprocal links in social networks. We define the key concepts as follows. A *heterogeneous and reciprocal link* is a link (u, v) that has its two nodes u and v belonging to different types, or groups (i.e., the nodes are heterogeneous) and its edge sign depends on the attitudes of both nodes (i.e., link establishment is reciprocal). *Heterogeneous and reciprocal social networks* are networks connected only by heterogeneous and reciprocal links. A *heterogeneous and reciprocal link prediction* problem (*ReHeLP*) is the prediction of links in heterogeneous and reciprocal social networks.

Heterogeneous and reciprocal social networks exist in many applications (e.g., online dating sites, education admission sites, as well as jobs, employment, career and recruitment sites). In online dating sites, we have: 1) users belonging to different groups (male or female); 2) links established only between users from different groups; and 3) link signs dependent on the compatibility of user pairs.

4 Methods

Given a directed graph $G = (V, E)$ with a sign (positive or negative) on each edge, we let $s(u, v)$ denote the sign of the edge (u, v) from node u to node v. That is, $s(u, v) = 1$ when the sign of $(u, v))$ is positive, 0 when negative. For different formulations of our task, we suppose that for a particular edge (u, v), the sign $s(u, v)$ is hidden and that we trying to infer it.

4.1 Feature Construction

The first step towards our heterogeneous and reciprocal link prediction is feature construction, which defines a collection of features for learning a model. The features are divided into two categories, according to their relationships to the entities in the networks.

Monadic Features. In social networks, the activity and popularity of an entity have impact on the behaviour of the entity. Therefore, the first category of characteristics of entities to be measured for link prediction is the activity and popularity of entities, which are the aggregated local relations of an entity to the rest of the world. This type of information represents the baseline, quantifying how many ingoing and outgoing edges a node could have.

The number of outgoing actions of a node measures how active an entity in the networks is, represented by its out-degree, the number of outgoing edges of a node in graph. We define the first monadic features based on the degree of the outgoing edges, as follows. An *outgoing edge* e of a node v is an edge that directs from v to another node. The *degree of outgoing edges* of a node $d_o(v)$ is

the number of outgoing edges from that node v. We also separate the outgoing edges according to their sign and define the degree of positive outgoing edges and the degree of negative outgoing edges, which represents not only the activity but also the general attitude of an entity to the world. The *degree of positive outgoing edges* of a node $d_o^+(v)$ is the number of outgoing edges from that node v and with positive sign. The *degree of negative outgoing edges* of a node $d_o^-(v)$, is the number of outgoing edges from that node v and with negative sign.

Similarly, the number of incoming actions of a node measures how popular an entity is in the network, represented by the degree of incoming edges. Therefore, we define the degree of positive incoming edges and the degree of negative incoming edges to model the popularity and again general attitude of an entity as follows. The *degree of positive incoming edges* of a node $d_i^+(v)$ is the number of incoming edges to that node v with positive sign. The *degree of negative incoming edges* of a node $d_i^-(v)$, is the number of incoming edges to that node v with negative sign.

The four monadic features $(d_o^+(v), d_o^-(v), d_i^+(v), d_i^+(v))$ will be used in our method to represent the activity and popularity as well as the general attitude of an entity in a network.

Dyadic Features. We also define dyadic features based on collaborative information. Collaborative information is the information extracted from a community that represents knowledge about the network derived from collaborative efforts. Collaborative information is the basis of collaborative filtering for recommendation, which makes automatic predictions about the interests of a user by collecting preferences or taste information from many other users. We make use of collaborative information for link prediction and extract dyadic features as in collaborative filtering.

The links of interest represent reciprocal relationships between entities, requiring *reciprocal* collaborative information to be considered [4,5,9]. Reciprocal collaborative information could be embedded in several different kinds of collaborative filtering frameworks. In [4], a general framework for reciprocal collaborative filtering was developed for recommendation, which was then extended in several variant methods [9]. Here, we contribute to integrating such collaborative information into a learnable framework for link prediction, rather than recommendation. Moreover, we aim at prediction of heterogeneous links, hence both nodes of a link cannot link to the same third node. For example, in people to people dating recommendation, a link only exists between a heterogeneous pair, i.e. a male type and a female type (we do not consider same-sex relationships in this work). In this bipartite representation the sender and recipient cannot both link to the same third person. Therefore, we consider a three step path involving both nodes within a potential link, which is defined as a *tetrad* in Definition 1.

Definition 1. *A tetrad $t(u, s_v, s_u, v)$ or $t(u, v)$ is a three step path among four different nodes $(u \to s_v \to s_u \to v)$ in a graph, where the source node u (sender) and a node similar to it s_u (defined by collaborative filtering) both belong to one of the two types, while the target node v (recipient) and another node similar to it s_v both belong to the other type.*

Table 1. Dyadic Features Based on Reciprocal Collaborative Information

Typical		Inverted				Transmissible	
RSR	SRS	RRS	RSS	SRR	SSR	SSS	RRR
$n(r_+s_+r_+)$	$n(s_+r_+s_+)$	$n(r_+r_+s_+)$	$n(r_+s_+s_+)$	$n(s_+r_+r_+)$	$n(s_+s_+r_+)$	$n(s_+s_+s_+)$	$n(r_+r_+r_+)$
$n(r_+s_+r_-)$	$n(s_+r_+s_-)$	$n(r_+r_+s_-)$	$n(r_+s_+s_-)$	$n(s_+r_+r_-)$	$n(s_+s_+r_-)$	$n(s_+s_+s_-)$	$n(r_+r_+r_-)$
$n(r_+s_-r_+)$	$n(s_+r_-s_+)$	$n(r_+r_-s_+)$	$n(r_+s_-s_+)$	$n(s_+r_-r_+)$	$n(s_+s_-r_+)$	$n(s_+s_-s_+)$	$n(r_+r_-r_+)$
$n(r_+s_-r_-)$	$n(s_+r_-s_-)$	$n(r_+r_-s_-)$	$n(r_+s_-s_-)$	$n(s_+r_-r_-)$	$n(s_+s_-r_-)$	$n(s_+s_-s_-)$	$n(r_+r_-r_-)$
$n(r_-s_+r_+)$	$n(s_-r_+s_+)$	$n(r_-r_+s_+)$	$n(r_-s_+s_+)$	$n(s_-r_+r_+)$	$n(s_-s_+r_+)$	$n(s_-s_+s_+)$	$n(r_-r_+r_+)$
$n(r_-s_+r_-)$	$n(s_-r_+s_-)$	$n(r_-r_+s_-)$	$n(r_-s_+s_-)$	$n(s_-r_+r_-)$	$n(s_-s_+r_-)$	$n(s_-s_+s_-)$	$n(r_-r_+r_-)$
$n(r_-s_-r_+)$	$n(s_-r_-s_+)$	$n(r_-r_-s_+)$	$n(r_-s_-s_+)$	$n(s_-r_-r_+)$	$n(s_-s_-r_+)$	$n(s_-s_-s_+)$	$n(r_-r_-r_+)$
$n(r_-s_-r_-)$	$n(s_-r_-s_-)$	$n(r_-r_-s_-)$	$n(r_-s_-s_-)$	$n(s_-r_-r_-)$	$n(s_-s_-r_-)$	$n(s_-s_-s_-)$	$n(r_-r_-r_-)$

A *tetrad* $t(u, v)$ captures a two step relationship across two types, which is the minimum indirect path between a pair of nodes (u, v). *Tetrad* is a novel type of sub-graph feature for heterogeneous and reciprocal link prediction, which captures collaborative information that cannot be captured by features used in existing link prediction approaches. The following feature sets for each pair of nodes (u, v) are then based on a variety of minimum indirect paths defined on the pair.

The first type of dyadic features is based on *Typical Reciprocal Collaborative Information*. As shown in Figure 2, this is the typical reciprocal collaborative filtering for people to people recommendation with two-way preferences [4,5]. In the figure, u_a is the source node (corresponding to u in Definition 1), u_r the target node (v in Definition 1), u_{str} and u_{sar} the similar nodes (s_v in Definition 1), u_{saa} and u_{sta} the similar nodes (s_u in Definition 1). From this configuration, we can construct two set of features. One set is (on the top half of the figure) to capture the collaborative information for predicting the recipient's preference. The other set is (on the bottom half of the figure) to capture the collaborative information for predicting the initiator's preference. Since we have positive or negative signs for each interaction and there are 3 interactions in each set as shown in the figure, we can create $2 * 2^3 = 16$ features of this type as in Table 1 indicated by RSR and SRS. In Table 1, a tetrad type is presented by the directions of three edges from u to v in a tetrad $t(u, v)$, where S and s mean a link from a precursor to a successor, R and r to a precursor from a successor, and their signs, where $+$ means a positive link and $-$ negative. n is used to represent the number of links of a tetrad type. To give an example, $n(r_+s_-r_+)$ means the total number of the type of tetrad $t(u, s_v, s_u, v)$ that have a positive link from u to s_v, a negative link to s_v from s_u and a positive link from s_u to v.

The second type of dyadic feature is based on *Inverted Reciprocal Collaborative Information*. This type of dyadic feature is derived by fitting the inverted collaborative filters [9] into the reciprocal collaborative filtering framework [4]. There are two types of inverted collaborative filters: user-based and item-based. In this work, we make use of both of these inverted collaborative filters to generate more features. We first take the recipient-based inverted collaborative filtering

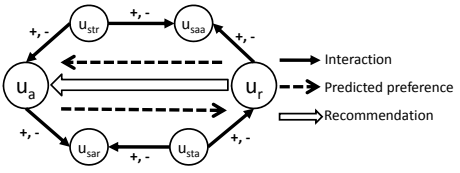


Fig. 2. Typical reciprocal collaborative filtering

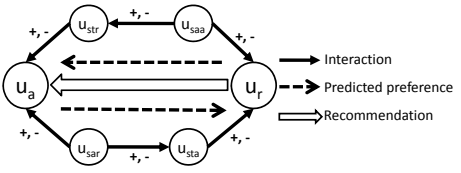


Fig. 3. Inverted reciprocal collaborative filtering with similar recipient

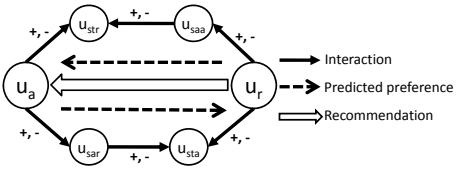


Fig. 4. Inverted reciprocal collaborative filtering with similar sender

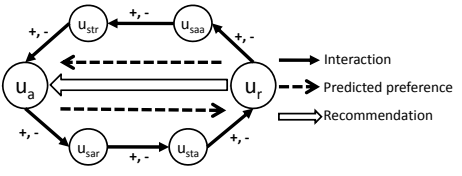


Fig. 5. Reciprocal collaborative filtering by preference transmission

and add another novel type of collaborative information (on the top half of the figure) to capture the preference of the recipient as shown in Figure 3. The original inverted collaborative filters only model positive signs for all interactions in the configuration. To allow more collaborative information to be considered by the learning system in our configuration, we use both positive and negative signs in any interaction within the configuration, and rely on the machine learning method to select discriminative features. Similarly, we add one more new type of collaborative information (on the top half of the figure) to the sender-based inverted collaborative filtering to capture the preference of the recipient as shown in Figure 4. We also allow both positive and negative signs for interaction. Features based on inverted collaborative information are summarised in Table 1 indicated by RRS, RSS, SRR and SSR.

The third type of dyadic feature is based on *Transmissible Collaborative Information*. Dyadic features are considered also to capture the transmissible properties of preferences as in Figure 5. Similar to [9], if we only consider positive interactions in creating the collaborative information, we should then have the property of preference transmission. This can be easily validated by the taste and attractiveness concept in [4]. Similarly, we have the first set of features (on the top half of the figure) to capture the recipient's preference and another set of features (on the bottom half of the figure) to capture the sender's preference. We again allow both positive and negative signs in any interaction within the configuration. Features based on preference transmission are illustrated in Table 1 indicated by SSS and RRR.

Each of these 64 tetrad types may provide different evidence about the sign of the edge from the initiator to the recipient, possibly some favouring a negative

sign and some favouring a positive sign. We encode this information in a 64-dimensional vector specifying the number of tetrads of each type that both nodes in a link are involved in. Notice that this is the first time a complete set of combinations showing all possible sources of collaborative information derived from the structure of tetrads has been used for link prediction.

4.2 Learning and Testing

To predict links, we first calculate the feature values and then calculate a measure of combined feature strength as the weighted combination of feature values, as follows:

$$s = \sum_{i=1}^{n} \omega_i x_i + \omega_0 \tag{1}$$

where s is the combined feature strength, x_i the value of the ith feature and ω_i the weight value for x_i. To learn the weights and convert this combined feature strength into an edge sign prediction, we use logistic regression, which will output a value in the range of $(0, 1)$ representing the probability of a positive edge sign:

$$p = \frac{1}{1 + e^{-s}} \tag{2}$$

where p is the predicted probability of an positive edge sign.

We will show in Section 5 that by using logistic regression, we are also able to uncover the contribution of each feature to the prediction by investigating the learned coefficients.

Once we have the learned model, testing is simply to calculate feature values for each test instance (pair of nodes) and input them into the learned model to compute the probability of a positive link between them. The instances are then classified into positive or negative according to the thresholding of the probability value with respect to a threshold.

5 Experiments

5.1 Setup

In these experiments, we aim to evaluate the proposed approach on link prediction of dating social networks in a real world dataset, which is a demanding real-world one. Link prediction on dating social networking is a typical heterogeneous and reciprocal link prediction problem, where the nodes are users and links are interactions. Here users are either of male or female type, links are only estimated between users of a different type and the link sign depends on the decisions of both users. The datasets were collected from a commercial social network (online dating) site containing interactions between users. Specifically, the data contains records, each of which represents a contact (communication)

Table 2. Dataset Description

	#Interaction	#Positive Link	#Negative Link	#User
Numbers in Dataset	1710332	264142(15%)	1446190(85%)	166699

by a tuple containing the identity of the contact's sender, the identity of the contact's recipient, and an indicator showing whether the interaction was successful (with a positive response from the recipient to the sender) or unsuccessful (with a negative response or no response).

The experiments were conducted on a dataset covering a four week period in March, 2010. The dataset contains all users with at least one contact in the specific period. The dataset used for this research is summarised in Table 2. We follow the methodology of Leskovec [12] and created a balanced dataset with equal numbers of positive and negative edges.

To the best of our knowledge, there is no existing work on edge sign prediction in heterogeneous and reciprocal networks. Therefore, we took the recent approach to positive and negative link prediction in online social networks by Leskovec et al. [12] as the baseline to compare to the proposed algorithm, which has been reported to significantly improve on previous approaches [12]. The baseline method uses all valid features except those based on two-step paths that are not valid for heterogeneous and reciprocal link prediction since the latter has a tetrad structure. Notice that although the baseline method has some utility in heterogeneous and reciprocal link sign prediction, it is not designed for that problem. To the best of our knowledge, we are the first to consider the heterogeneous and reciprocal link sign prediction problem.

We use accuracy, precision and recall as evaluation metrics for evaluation of the proposed algorithm. We also use the receiver operating characteristic (ROC) and the area under the ROC curve (AUC) for our evaluation.

Algorithms for feature extraction were implemented using SQL in Oracle 11. Learning and testing algorithms were implemented in Matlab. For the large scale dataset in Table 2, feature extraction required less than 1 hour. Training on 90% of the balanced dataset and testing on the remaining 10% of the dataset took about 1 minute on a workstation with 64-bit Windows 7 Professional, 2 processors of Intel(R) Xenon(R) CPU x5660@2.80GHz and 32GB RAM.

5.2 Results

To compare the proposed method to the baseline method, we conducted experiments to generate values for the evaluation metrics from two methods and statistically tested significance of differences using a paired t-test.

We used 10 fold cross-validation (CV) repeated 10 times and hold-out to generate results for evaluation. For hold-out, we hold 20% of the data for testing and train the model using the remaining 80% of the data. The comparative results are shown in Figure 6 with details in Table 3. The proposed method achieves about 78% predictive accuracy on average on 100 runs by 10 fold CV repeated

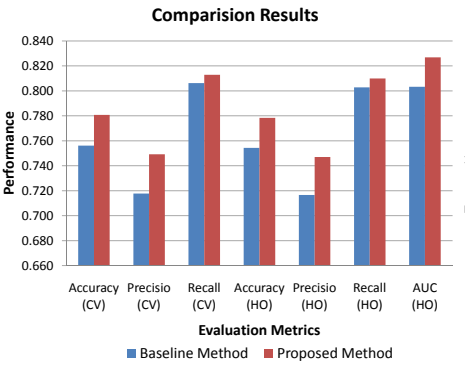


Fig. 6. Comparative Results. CV indicates 10-fold CV and HO holding out.

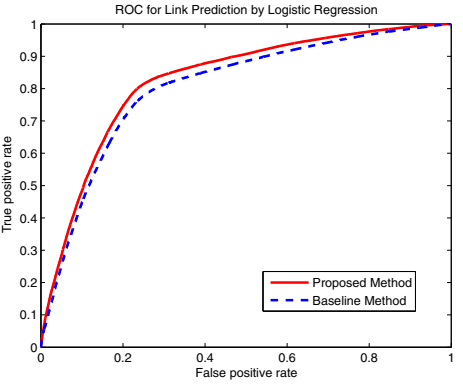


Fig. 7. ROC for comparison of the proposed method and baseline method

Table 3. Comparative Results. ("*" indicates improvement of the method.)

		Baseline	**ReHeLP**	Improvement
10-fold CV repeated 10 times	Accuracy	0.756	**0.781***	2.46%
	Precision	0.718	**0.749***	3.14%
	Recall	0.806	**0.813***	0.67%
20% hold out for testing	Accuracy	0.754	**0.778***	2.40%
	Precision	0.717	**0.747***	3.04%
	Recall	0.803	**0.810***	0.71%
	AUC	0.803	**0.827***	2.35%

10 times while the baseline method only has less than 76% predictive accuracy on average, showing the proposed method outperforms the baseline method by about 2.5% predictive accuracy. For hold-out evaluation, the proposed method similarly outperforms the baseline method by 2.4% predictive accuracy. The proposed method also outperforms the baseline method in terms of precision, recall and AUC as shown in Table 3, where the proposed method achieved 3.14%, 0.67% improvement over the baseline method for precision and recall respectively by 10 fold cross-validation repeated 10 times, and 3.04%, 0.71% improvement over the baseline method for precision and recall respectively by hold-out. The threshold selected for this evaluation is based on the optimal operating point selected using the ROC curve in Figure 7, where the ROC curve of the proposed method remains above that of the baseline method also indicating the improvement by the former. On the AUC of the ROC curve, the proposed method outperforms the baseline method by 2.35%.

A paired t-test is used to assess whether the means of the results of our method and the compared method are statistically different from each other. The result of a paired t-test on the corresponding predictive accuracy x of the proposed method and predictive accuracy y of the compared baseline method by

Table 4. Paired t-test at the 5% Significance Level

Hypothesis (h)	p-value	95% Confidence Interval
1	2.74E-98	[0.0241 0.0251]

10-fold cross-validation repeated 10 times is shown in Table 4. Here the paired t-test tests the null hypothesis that data in the difference $x - y$ are a random sample from a normal distribution with mean 0 and unknown variance, against the alternative that the mean is not 0, i.e. the null hypothesis that the results come from populations with equal means, against the alternative that the means are unequal. Our experiments show that the test rejects the null hypothesis at the $\alpha = 0.05$ significance level as shown by the hypothesis $h = 1$ in the table. Notice that the 95% confidence interval on the difference mean contains a positive interval that does not contain 0, which indicates that the mean of the predictive accuracy of the proposed method is greater than that of the baseline method. Moreover, the p value has fallen below $\alpha = 0.05$ and in fact even below $\alpha = 0.01$, which can be considered as a very significant difference (significant improvement in accuracy).

6 Conclusion

We have presented a learning framework for the heterogeneous and reciprocal link prediction problem. To the best of our knowledge, this the first work to address the link sign prediction problem in heterogeneous and reciprocal social networks. The improvement gained by the proposed approach has been clearly demonstrated by a set of extensive experiments. The experiments were conducted in demanding real world datasets collected from a commercial social network site, which shows that the proposed method is able to make heterogeneous and reciprocal link predictions and outperforms the use of existing link prediction techniques.

Future work will include investigating methods to use better collaborative information and understand the way that the collaborative information contributes to the prediction in order to design and make use of improved features.

Acknowlegement. This project is funded by the Smart Services CRC under the Australian Government's Cooperative Research Centre program. We would also like to thank our industry partners for providing the datasets.

References

1. Acar, E., Dunlavy, D.M., Kolda, T.G.: Link prediction on evolving data using matrix and tensor factorizations. In: Proceedings of the 9th IEEE International Conference on Data Mining Workshops, pp. 262–269 (2009)

2. Bilgic, M., Namata, G.M., Getoor, L.: Combining collective classification and link prediction. In: Proceedings of the 7th IEEE International Conference on Data Mining Workshops, pp. 381–386 (2007)
3. Brzozowski, M.J., Hogg, T., Szabo, G.: Friends and foes: ideological social networking. In: Proceeding of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems, pp. 817–820 (2008)
4. Cai, X., Bain, M., Krzywicki, A., Wobcke, W., Kim, Y.S., Compton, P., Mahidadia, A.: Collaborative Filtering for People to People Recommendation in Social Networks. In: Li, J. (ed.) AI 2010. LNCS, vol. 6464, pp. 476–485. Springer, Heidelberg (2010)
5. Cai, X., Bain, M., Krzywicki, A., Wobcke, W., Kim, Y.S., Compton, P., Mahidadia, A.: Learning collaborative filtering and its application to people to people recommendation in social networks. In: Proceedings of the 10th IEEE International Conference on Data Mining, pp. 743–748 (2010)
6. Cao, B., Liu, N.N., Yang, Q.: Transfer learning for collective link prediction in multiple heterogenous domains. In: Proceedings of the 27th International Conference on Machine Learning, pp. 159–166 (2010)
7. Chao, W., Satuluri, V., Parthasarathy, S.: Local probabilistic models for link prediction. In: Proceedings of the 7th IEEE International Conference on Data Mining, pp. 322–331 (2007)
8. Hasan, M.A., Chaoji, V., Salem, S., Zaki, M.: Link prediction using supervised learning. In: Workshop on Link Analysis, Counter-terrorism and Security (at SIAM Data Mining Conference) (2006)
9. Krzywicki, A., Wobcke, W., Cai, X., Mahidadia, A., Bain, M., Compton, P., Kim, Y.S.: Interaction-Based Collaborative Filtering Methods for Recommendation in Online Dating. In: Chen, L., Triantafillou, P., Suel, T. (eds.) WISE 2010. LNCS, vol. 6488, pp. 342–356. Springer, Heidelberg (2010)
10. Kunegis, J., Lommatzsch, A., Bauckhage, C.: The slashdot zoo: mining a social network with negative edges. In: Proceedings of the 18th International Conference on World Wide Web, pp. 741–750 (2009)
11. Lampe, C.A., Johnston, E., Resnick, P.: Follow the reader: filtering comments on slashdot. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1253–1262 (2007)
12. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Predicting positive and negative links in online social networks. In: Proceedings of the 19th International Conference on World Wide Web, pp. 641–650 (2010)
13. Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. In: Proceedings of the 12th International Conference on Information and Knowledge Management, pp. 556–559 (2003)
14. Lu, Z., Savas, B., Tang, W., Dhillon, I.S.: Supervised link prediction using multiple sources. In: Proceedings of the 10th IEEE International Conference on Data Mining, pp. 923–928 (2010)

Detecting Multiple Stochastic Network Motifs in Network Data

Kai Liu, William K. Cheung, and Jiming Liu

Department of Computer Science
Hong Kong Baptist University
224 Waterloo Road, Kowloon Tong, Hong Kong
{kliu,william,jiming}@comp.hkbu.edu.hk

Abstract. Network motif detection methods are known to be important for studying the structural properties embedded in network data. Extending them to stochastic ones help capture the interaction uncertainties in stochastic networks. In this paper, we propose a finite mixture model to detect multiple stochastic motifs in network data with the conjecture that interactions to be modeled in the motifs are of stochastic nature. Component-wise Expectation Maximization algorithm is employed so that both the optimal number of motifs and the parameters of their corresponding probabilistic models can be estimated. For evaluating the effectiveness of the algorithm, we applied the stochastic motif detection algorithm to both synthetic and benchmark datasets. Also, we discuss how the obtained stochastic motifs could help the domain experts to gain better insights on the over-represented patterns in the network data.

Keywords: Stochastic motifs, finite mixture models, expectation maximization algorithm, social networks.

1 Introduction

Network motifs, also known as simple building blocks of complex networks, are defined as patterns of interactions that appear in different parts of a network more frequently than those found in randomized networks. With the network represented as a graph, network motifs can be interpreted as the over-represented subgraph patterns embedded in the graph. Since the pioneering work by Shen-Orr *et. al* [1], there have been a lot of research works on detecting network motifs in biological networks [4,5,6] with the objective to gain insights on the relationship between the network structural properties and the functions they possess. Milo *et al.* [2,3] generalized the idea to characterize a broad range of networks, including ecosystem food webs, neuronal networks, World Wide Web, etc. Recently, network motif detection has also been applied to social network analysis. For example, an email based social network can be well characterized by the Z-score distribution of embedded 3-node subgraph patterns [8,9].

Most of the existing works on network motif detection assume that the network motif is deterministic, which means that the corresponding subgraph

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 205–217, 2012.

patterns either appear completely or are missing totally. Deterministic network motif detection methods could give inaccurate results if the motifs exhibit stochastic properties. The corresponding stochastic network can be modeled as a mixture of a background random ensemble and families of mutually similar but not necessarily identical interconnection patterns represented by a stochastic network motif [6] (which is also called probabilistic motif in [5]). Stochastic motif detection can then be casted as a missing-value inference and parameter estimation problem under a Bayesian framework. Expectation-Maximization(EM) algorithm and Gibbs sampling can readily be adopted [6,10].

Recently, Liu *et al.* [11] applied the finite mixture model to analyze social media but with the assumption that there is only one stochastic motif. This paper generalizes this work to model stochastic network as a finite mixture model with k components ($k \geq 1$) and adopt the Bayesian approach for detecting the optimal set of multiple stochastic motifs. The paper is organized as follow. Section 2 presents the problem formulation. Evaluation results obtained via experiments performed based on both synthesis and benchmark datasets are reported in Section 3. Section 4 concludes the paper with future research directions.

2 Network Motif Analysis in Social Media

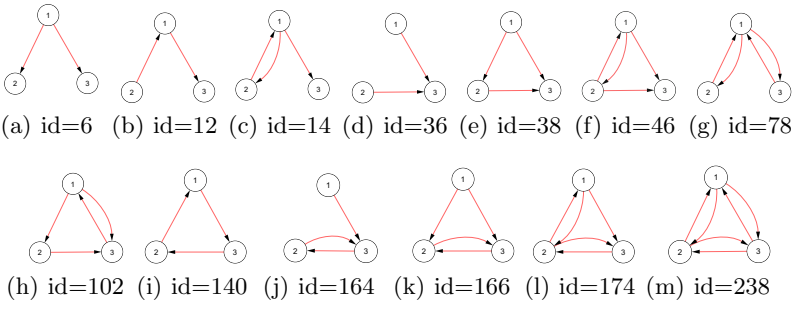
Analyzing triads embedded in networks have long been found important in conventional social network analysis. However, local interaction patterns (or termed as "ties" in social network analysis community) which are salient for characterizing the overall structure of the networks could appear with stochastic variations. It makes conventional motif detection methods problematic as demonstrated in [11]. For large online networks which contain interactions of millions of different individual entities, the incorporation of stochastic models becomes especially essential for more robust motif detection. This is analogous to the need of hidden Markov Model (HMM) for more robust speech recognition and that of conditional random field (CRF) for information extraction. For stochastic motif detection, the target of detection is the embedded network motifs (foreground) and the other links are modeled as the random background.

Relationships or interactions among N elementary units in a population could be represented as a graph G with N nodes and a set of edges denoted by an adjacency matrix $\boldsymbol{A} = (a_{ij})_{N \times N}$. For directed graphs, $a_{ij} = 1$ if there is a directed edge pointing from node i to node j, and 0 otherwise. For undirected graphs, $a_{ij} = 1$ if node i and node j are connected, and 0 otherwise. Subsets of nodes in G with only the local connectivity considered define subgraphs of G. A subgraph S with n nodes can be described by an adjacency matrix $X_S = (x_{ij})_{n \times n}$, where x_{ij} is either 0 or 1 to indicate its connectivity. By sampling subgraphs of a relatively small size (say, triads) from G, the frequency distribution of their appearance can characterize the local structural properties of the graph. To extend from this, a set of subgraphs with "structurally similar" adjacency matrices defines a stochastic network subgraph pattern which if over-represented defines a stochastic network motif M.

2.1 Canonical Forms of Subgraphs for Modeling Stochastic Motifs

Enumerating or sampling subgraphs from a network is the pre-processing step needed before related stochastic models can be applied. A well-known problem is the handling of subgraph isomorphism. Intuitive speaking, structurally equivalent subgraph instances could have their nodes labeled in different orders, making those equivalent subgraphs associated with very different adjacency matrices. Identifying subgraph isomorphism itself is NP-complete in general [13]. Some heuristic computational tricks could be applied to reduce the computational complexity issue on average. In this work, an efficient graph/subgraph isomorphism testing algorithm Nauty [12] is used to check for structurelly equivalent subgraphs and relabel them based on a canonical one so that their appearances can be well aggregated and the stochastic model learning can be accurate.

The remaining question is the choice of the canonical forms. Existing methods for detecting deterministic motifs assume that the motifs are independent and the choices of the canonical forms for the isomorphically equivalent groups of subgraph instances can just be independently considered. However for stochastic motifs, one should expect a stochastic motif model which gives a high probability value to the canonical form of a subgraph pattern A should give also a relatively high value to that of a subgraph pattern B which is a subgraph of A. In other words, the canonical forms for the different isomorphically equivalent subgraph patterns should be chosen in such a way that one being the subset of another should be "aligned" as reflected in their node labeling orders. With this considered, we carefully derived the set of canonical subgraph patterns for subgraphs with 3 nodes as shown in Figure 1. For subgraphs with more than 3 nodes, we are currently studying the possibility of building the corresponding canonical forms efficiently by joining and/or extending the canonical adjacency matrices of 3-node subgraphs [14].



(a) id=6 (b) id=12 (c) id=14 (d) id=36 (e) id=38 (f) id=46 (g) id=78

(h) id=102 (i) id=140 (j) id=164 (k) id=166 (l) id=174 (m) id=238

Fig. 1. All possible subgraphs of canonical form with 3 nodes

2.2 Finite Mixture Model

With the assumption that a stochastic network can be modeled as a mixture of families of independent foreground stochastic motifs embedded in a background random ensemble, each subgrah in the stochastic network can be regarded as either generated from the background or from one of the foreground motifs. In

this paper, we extend from the mixture model in [6,11] that multiple stochastic motifs can be detected and the number of motifs required can be estimated.

Assuming there exist k stochastic motifs $\boldsymbol{M}_f = \{M_1, \cdots, M_k\}$ which are represented as a set of probability matrices $\boldsymbol{\Theta}_f = \{\Theta_1, \cdots, \Theta_k\}$, with $\Theta_h = (\theta_{ij}^h)_{n\times n}, 0 \leq \theta_{ij} \leq 1, 1 \leq h \leq k$. θ_{ij}^h denotes the probability that there is an edge from node i to j in the h-th motif. The background ensemble M_0 is characterized by a family of randomized networks generated from a given stochastic network which contain the same number of nodes and edges, and the same statistics for the nodes' in/out degrees.

Moreover, let $\{S_1, \cdots, S_W\}$ denote a set of subgraph instances sampled from a given network, $\boldsymbol{X} = \{\boldsymbol{X}^1, \cdots, \boldsymbol{X}^W\}$ denote the adjacency matrices corresponding to the subgraph instances (observed data) where $\boldsymbol{X}^w = (x_{ij}^w)_{n\times n}$ and $x_{ij}^w = \{0, 1\}$, Z_h^w denotes an indicator variable taking the value of 1 if subgraph instance S_w comes from the model M_h or 0 otherwise, and thus $\boldsymbol{Z} = (\boldsymbol{Z}^1, \cdots, \boldsymbol{Z}^W)^T$ form the missing data of the problem, where $\boldsymbol{Z}^w = (Z_0^w, \cdots, Z_k^w)^T$. The probability that $\boldsymbol{X}^w$ comes from M_h is given as

$$p(\boldsymbol{X}^w|\Theta_h) = \prod_{i=1}^{n}\prod_{j=1}^{n}(\theta_{ij}^h)^{x_{ij}^w}(1-\theta_{ij}^h)^{1-x_{ij}^w}. \tag{1}$$

Also, let $\boldsymbol{\lambda} = (\lambda_0, \cdots, \lambda_k)$ be the mixing portion of the mixture model which also denotes the prior probabilities of $\Pr(Z^w = 1), w = \{1, \cdots, W\}$.

The stochastic motif detection problem can thus be casted as a maximum likelihood estimation problem for $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_f, \boldsymbol{\lambda}\}$ where the log-likelihood function for the complete data is given as

$$l(\boldsymbol{\Theta}) = \log p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\Theta}) \quad = \sum_{w=1}^{W}\sum_{h=0}^{k} Z_h^w \log \lambda_h + \sum_{w=1}^{W}\sum_{h=0}^{k} Z_h^w \log p(\boldsymbol{X}^w|\Theta_h). \tag{2}$$

The EM algorithms for estimating $\boldsymbol{\Theta}_f$ and $\boldsymbol{\lambda}$ will be presented in the next section.

For the background model, we are interested in the probability of observing the subgraph instance S_w in the background model $p(\boldsymbol{X}^w|\Theta_0)$ instead of Θ_f. As in [6,11], the background model is estimated by counting the subgraph instances in randomized networks. We first generate a set of randomized networks. For each randomized network described by an adjacency matrix $\boldsymbol{A} = (a_{ij})_{N\times N}$, we randomly choose pairs of connections and repeatedly swap the target of them until the network is well randomized, while keeping the incoming and outgoing degrees of each node remain unchanged, i.e., keeping the summation of each row and each column in the adjacency matrix unchanged. Subgraphs are then sampled from the randomized networks. $p(\boldsymbol{X}^w|\Theta_0)$ is estimated as N_w/N_{total}, where N_w is the number of the subgraph S_w sampled from the ensemble of the randomized networks and N is the total number of subgraphs sampled with the same size with S_w.

2.3 Basic EM Algorithm

For learning probabilistic models with missing data (unknown motifs for our case), the Expectation-Maximization (EM) algorithm [15] is typically used for obtaining the Maximum Likelihood (ML) estimates of the model parameters.

The EM algorithm produces a sequence of estimates by alternatingly applying the E-step and M-step until it converges.

– E-step: Compute the complete data expectation of log-likelihood $E[l(\boldsymbol{\Theta})]$ given the observed data $\boldsymbol{X}$ and the current estimates of model parameters $\hat{\boldsymbol{\Theta}}$. We have

$$E[l(\boldsymbol{\Theta})] = \sum_{w=1}^{W}\sum_{h=0}^{k} E[Z_h^w]\log\hat{\lambda}_h + \sum_{w=1}^{W}\sum_{h=0}^{k} E[Z_h^w]\log p(\boldsymbol{X}^w|\hat{\Theta}_h), \quad (3)$$

where

$$E[Z_h^w] = E[Z_h^w|\boldsymbol{X},\hat{\boldsymbol{\Theta}}] = \frac{p(\boldsymbol{X}^w|\hat{\Theta}_h)\hat{\lambda}_h}{\sum_{j=0}^{k} p(\boldsymbol{X}^w|\hat{\Theta}_j)\hat{\lambda}_j} \quad (4)$$

– M-step: The model parameters are estimated by maximizing the expectation of the log-likelihood, given as

$$(\boldsymbol{\lambda}^*, \boldsymbol{\Theta}_f^*) = \arg\max_{\boldsymbol{\lambda},\boldsymbol{\Theta}_f} E[l(\boldsymbol{\Theta})]. \quad (5)$$

And the updating rules for $\boldsymbol{\lambda}$ and Θ_h are given as

$$\lambda_h^* = \frac{1}{W}\sum_{w=1}^{W} E[Z_h^w] \qquad \lambda_0^* = 1 - \sum_{h=1}^{k}\lambda_h^* \quad (6)$$

$$(\theta_{ij}^h)^* = \frac{\hat{\alpha}_{ij}^h}{\hat{\alpha}_{ij}^h + \hat{\beta}_{ij}^h}, \quad \hat{\alpha}_{ij}^h = \sum_{w=1}^{W} E[Z_h^w]x_{ij}^w, \quad \hat{\beta}_{ij}^h = \sum_{w=1}^{W} E[Z_h^w](1 - x_{ij}^w). \quad (7)$$

Note that $p(\boldsymbol{X}^w|\Theta_0)$ is estimated based on the subgraph statistics in the ensemble of randomized networks as explained in the previous section.

2.4 Learning the Optimal Number of Motifs

To determine the optimal number of stochastic motifs automatically, we adopt the *Component-wise EM for Mixture*(CEM^2) which was proposed to integrate both the model parameter estimation and model selection steps into one single EM algorithm [16]. The general idea of CEM^2 is to update the parameters of each component one by one so that the component with very low support by the data can be pruned. CEM^2 starts from all possible k-component mixtures and prunes the died components($\lambda_h = 0$) sequentially at each EM iteration.

CEM^2 implements the *minimum message length*(MML) criterion [17] to select the number of components. The best parameter estimate for the mixture model is the one minimizing the message length $L[\boldsymbol{\Theta}, \boldsymbol{X}]$, which is given by

$$L[\boldsymbol{\Theta}, \boldsymbol{X}] = L[\boldsymbol{\Theta}] + L[\boldsymbol{X}|\boldsymbol{\Theta}], \quad (8)$$

where $L[\boldsymbol{\Theta}]$ is the minimum message length for prior information, and $L[\boldsymbol{X}|\boldsymbol{\Theta}]$ is the minimum message length for data which can be estimated as $-\log p(\boldsymbol{X}|\boldsymbol{\Theta})$. As in [16], the final cost function (message length) $L[\boldsymbol{\Theta}, \boldsymbol{X}]$ is given by

$$L[\boldsymbol{\Theta}, \boldsymbol{X}] = \frac{N}{2}\sum_{m:\lambda_h>0}\log(\frac{W\lambda_h}{12}) + \frac{k_{nz}}{2}\log\frac{W}{12} + \frac{k_{nz}(N+1)}{2} - \log p(\boldsymbol{X}|\boldsymbol{\Theta}), \quad (9)$$

where k_{nz} is the number of components with non-zero probability, and N is the number of parameters specifying each component. The detailed steps of CEM^2 to motif detection can be found in Algorithm 1.

Algorithm 1. CEM^2 Algorithm

Input: Subgraphs $\boldsymbol{X} = \{\boldsymbol{X}_1, \cdots, \boldsymbol{X}_W\}$, ϵ, k_{min}, k_{max}, initial parameters $\hat{\boldsymbol{\Theta}}(0) = \{\hat{\Theta}_1, \cdots, \hat{\Theta}_{k_{max}}; \hat{\lambda}_1, \cdots, \hat{\lambda}_{k_{max}}\}$
Output: Mixture model with optimal $\boldsymbol{\Theta}^*$

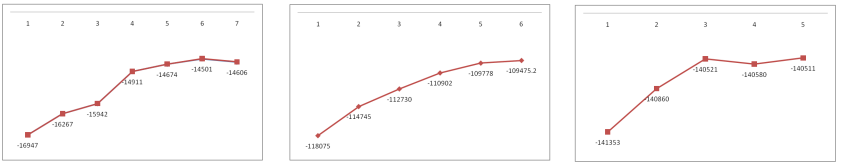
$t \leftarrow 0, k_{nz} \leftarrow k_{max}, L_{min} \leftarrow +\infty$
$u_h^w \leftarrow p(\boldsymbol{X}^w|\hat{\Theta}_h), c_h \leftarrow \max\{0, (\sum_{w=1}^{W} E[Z_h^w]) - \frac{N}{2}\}$, for $h = 1, \cdots, k_{max}$, and $w = 1, \cdots, W$
while $k_{nz} \geq k_{min}$ **do**
  repeat
    $t \leftarrow t + 1$
    for $h = 1$ to k_{max} **do**
      E-step: $E[Z_h^w] = u_h^w \hat{\lambda}_h (\sum_{j=0}^{k_{max}} u_j^w \hat{\lambda}_j)^{-1}$, $\lambda_h \leftarrow c_h (\sum_{j=0}^{k_{max}} c_j)^{-1}$
      M-step: $\{\hat{\lambda}_1, \cdots, \hat{\lambda}_{k_{max}}\} \leftarrow \{\hat{\lambda}_1, \cdots, \hat{\lambda}_{k_{max}}\} (\sum_{h=0}^{k_{max}} \hat{\lambda}_h)^{-1}$
        $\hat{\lambda}_0 = 1 - \sum_{h=1}^{k_{max}} \hat{\lambda}_h$
        if $\hat{\lambda}_h > 0$ **then**
          update $\boldsymbol{\Theta}_f$ according to Eq.(7), and $u_h^w \leftarrow p(\boldsymbol{X}^w|\hat{\Theta}_h)$
        else
          $k_{nz} \leftarrow k_{nz} - 1$
        end if
    end for
    $\hat{\boldsymbol{\Theta}}(t) = \{\hat{\Theta}_1, \cdots, \hat{\Theta}_{k_{max}}; \hat{\lambda}_0, \cdots, \hat{\lambda}_{k_{max}}\}$
    calculate $L[\hat{\boldsymbol{\Theta}}(t), \boldsymbol{X}]$ according to Eq.(9)
  until $L[\hat{\boldsymbol{\Theta}}(t-1), \boldsymbol{X}] - L[\hat{\boldsymbol{\Theta}}(t), \boldsymbol{X}] < \epsilon |L[\hat{\boldsymbol{\Theta}}(t-1), \boldsymbol{X}]|$
  if $L[\hat{\boldsymbol{\Theta}}(t-1), \boldsymbol{X}] \leq L_{min}$ **then**
    $L_{min} \leftarrow L[\hat{\boldsymbol{\Theta}}(t-1), \boldsymbol{X}]$
    $\boldsymbol{\Theta}^* \leftarrow \hat{\boldsymbol{\Theta}}(t)$
  end if
  $h^* \leftarrow \arg\min_h \{\hat{\lambda}_h > 0\}$, $\hat{\lambda}_h \leftarrow 0$, $k_{nz} \leftarrow k_{nz} - 1$
end while

3 Experimental Results

In this section, we present experimental results to demonstrate first the correctness of the detected stochastic motifs using synthetic datasets. Then, we further present the results of applying the stochastic motif detection algorithm to some real datasets and provide interpretation of the results obtained.

3.1 Results on Synthetic Networks

We generated a set of synthesized networks for correctness evaluation. Each network is generated by 1) creating a group of subgraphs coming from a known set of reference stochastic motifs as foreground models and 2) adding random links among the subgraphs to generate random background. In particular, we chose the subgraphs commonly found in many real networks, e.g., id is 38, 46, 166, 174, and 238 (see Fig. 1) as the reference motifs. We then applied our method to the synthetic networks we generated. In order to avoid the EM algorithm being



(a) Synthetic data (b) Amazon network (c) Wiki-Vote network

(d) Slashdot network (e) Epinions network

Fig. 2. The plot of the expected log likelihood under different numbers of motifs

trapped into local optima, we ran the EM algorithm several times with different initializations to report the best $\boldsymbol{\Theta}$ in terms of the likelihood value.

Figs. 3(a) - 3(e) show the stochastic motifs obtained by the multiple motif detection method in the synthetic networks. According to Fig. 2(a), the value of $E[l(\boldsymbol{\Theta})]$ increases a lot when the motif number varies from 1 to 5. There is a sharp drop in the increasing rate for the value of $\mathrm{E}[l(\boldsymbol{\Theta})]$ when $k = 5, 6, 7$. This is consistent to the fact that there are 5 reference motifs used for generating the synthetic networks. A similar conclusion can be drawn by referring to Table 1.

Table 1. λs when the number of motifs is 5, 6 and 7 ($\times 10^{-2}$)

	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7
$k=5$	7.1	9.8	8.6	7.7	**2**	-	-
$k=6$	3.8	7.9	8.5	10.1	**13.5**	0.2	-
$k=7$	9.1	9.5	9.1	6.0	**9.6**	0.1	0.1

Table 2. Dataset statistics ($\times 10^3$)

	Amazon	Wiki	Slash	Epinions
# nodes	262	8	77	76
# edges	1,235	104	828	509
# subgraphs	7,685	13,329	67,361	70,911

3.2 Results on Benchmark Datasets

We have also applied the stochastic motif detection algorithm to large-scale social network datasets named "Amazon", "Wiki-Vote", "Slashdot" and "Epinions" which are obtained as described in [18,19]. The dataset "Amazon" considers the Customers Who Bought This Item Also Bought feature of the Amazon website. If a product A is frequently co-purchased with product B, the graph contains an directed edge from node A to node B. "Wiki-Vote" is a network consisting of voting interaction for Wikipedia admin candidates. The link refers to a vote from a user to an admin candidate represented a user agree or disagree the promotion of the admin candidate. "Slashdot" is a social network of technology blog. The links in this network are the designations of "friends" or "foes". "Epinions" is a trust network, where we can know the trust or distrust relations of the users from the directed links between each other. Table 2 lists the statistics of these

four datasets. These networks have order of tens to hundreds of thousands of nodes and hundreds of thousands to millions of edges. In each network, we know the directions of all the edges.

Figs. 2(b) - 2(e) show the expected maximum log likelihood values E[$l(\boldsymbol{\Theta})$] of the mixture models with different component numbers in the four datasets. Here we can determine the best number of motifs by visual inspection to identify the points where the increase of E[$l(\boldsymbol{\Theta})$] starts to slow down. Also, by referring to tables 3 - 6, the values of λ with different motif numbers in these four datasets also hint us the best optimal number to choose from (as marked with bold face in the tables). For instance, for Amazon network, when the number of motifs k is set to 6, the value of λ_6 is very small. This hints that the 6-th motif is only "supported" by a very limited number of subgraph instances and thus 5 stochastic motifs could be enough. Similar results were obtained for Wiki-Vote, Slashdot and Epinions networks.

Table 3. λs for Amazon ($\times 10^{-2}$)

	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
$k=4$	2.3	3.5	2.1	1.5	-	-
$k=5$	1.7	1.6	2.3	2.3	**1.6**	-
$k=6$	2.2	2.7	1.4	1.3	**1.8**	0.1

Table 4. λs for Wiki-Vote ($\times 10^{-2}$)

	λ_1	λ_2	λ_3	λ_4
$k=2$	1.0	1.2	-	-
$k=3$	2.0	1.1	**1.1**	-
$k=4$	1.2	3.9	**4.7**	0.2

Table 5. λs for Slashdot ($\times 10^{-3}$)

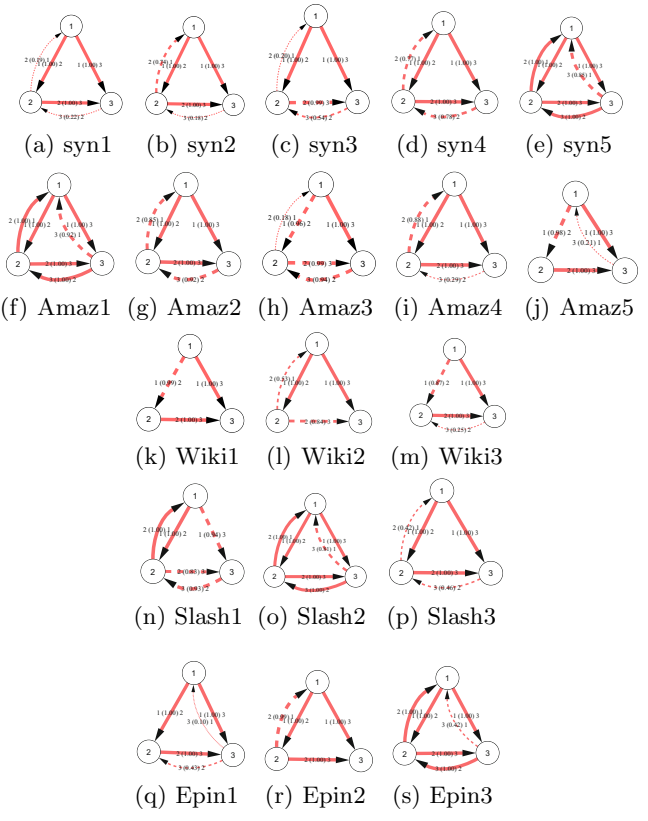
	λ_1	λ_2	λ_3	λ_4	λ_5
$k=3$	0.8	1.3	**2.9**	-	-
$k=4$	1.5	1.8	**1.6**	0.42	-
$k=5$	1.1	2.2	**1.4**	0.32	0.39

Table 6. λs for Epinions ($\times 10^{-3}$)

	λ_1	λ_2	λ_3	λ_4	λ_5
$k=3$	8.4	4.7	**8.0**	-	-
$k=4$	8.1	5.7	**2.3**	0.45	-
$k=5$	9.2	4.8	**5.8**	0.54	0.27

Fig. 3 shows the stochastic motifs detected in the datasets we used. Similar to [11], one can make interpretations on the networks of study based on the motifs extracted, which can in turn be validated by related domain experts. For instance, we made the following observations which seems revealing some local structural properties of the networks:

– By referring to the results obtained based on the Amazon dataset (Figs. 3(f) - 3(j)), we observed the following patterns: i) a 3-node pattern (Fig. 3(f)) where three products are always co-purchased bidirectionally; ii) a 3-node pattern (Fig. 3(g)) where only two pairs of products are co-purchased bidirectionally but not the third pair; iii) some other 3-node patterns where only one pair of products are co-purchased bidirectionally but not the other two (Figs. 3(h),3(i)); and iv) a 3-node pattern where the co-purchasing is never done directionally for the related products. It could be interesting to further analyze whether the four patterns are corresponding to different product characteristics, which could in turn result in some more context specific product recommendation methodologies.



(a) syn1 (b) syn2 (c) syn3 (d) syn4 (e) syn5

(f) Amaz1 (g) Amaz2 (h) Amaz3 (i) Amaz4 (j) Amaz5

(k) Wiki1 (l) Wiki2 (m) Wiki3

(n) Slash1 (o) Slash2 (p) Slash3

(q) Epin1 (r) Epin2 (s) Epin3

Fig. 3. Stochastic motifs detected in different datasets with the edge width showing the corresponding probability of edge appearance, and the edge label gives the actual probabilities. E.g., 1(0.92)2 means the occurrence probability of edge x_{12} is 0.92. A dashed edge here implies a probability value less than 1.

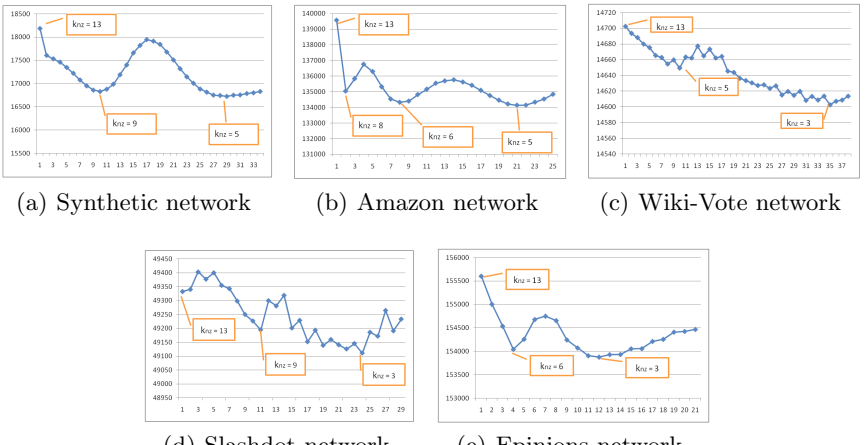
- From the results obtained based on the Wiki-Vote dataset (Figs. 3(k) - 3(m)), it is also interesting to observe the following patterns: i) a 3-node pattern (Fig. 3(k)) where co-voting never occurs; and ii) some 3-node patterns where co-voting only occasionally occurs for one pair of voters but not the other pairs (Figs. 3(l) and 3(m)). In general, co-voting activities within a triad are not commonly observed. We believe that this could be related to the user psychology behind the voting process, requiring again further investigation effort with respect to the corresponding application context.
- All the motifs detected in these social networks consist of a basic feed-forward loop structure (structure of Fig. 1(e)) with some additional edges. The feed-forward loop structure is the most popular deterministic motif found in biological and social networks, which follows the status theory in social networks proposed in [18]. E.g., if A regards B as having higher status (a link from A to B), and B regards C as having higher status (a link from B to C), so A should regard C as having higher status and hence be inclined to link

from A to C. So, the feed-forward loop structure is often over-represented while the feedback loop (structure of Fig. 1(i) having link from C to A) is under-represented instead.

As inspired by [18], we plan to make reference to different social psychology theories developed in social science to validate and gain further insights and thus explanation on the underlying social behaviors embedded in the social media.

3.3 Effectiveness of CEM^2 in Estimating Optimal Number of Motifs

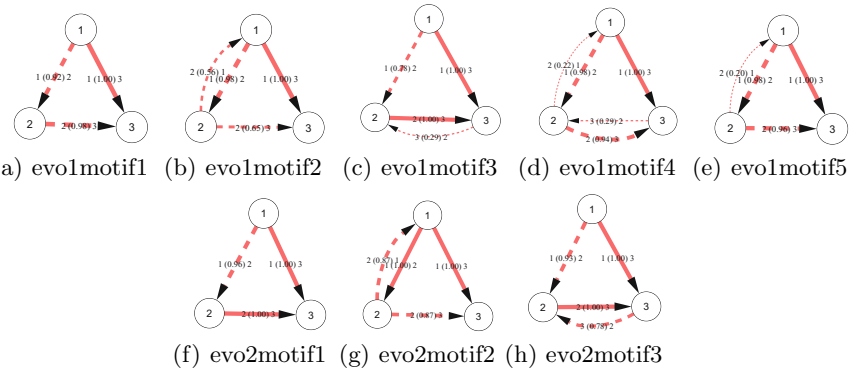
Fig. 4 shows how the cost functions $L(\boldsymbol{\Theta}, \boldsymbol{X})$ evolve throughout the CEM^2 iterations. Starting from the maximum possible number of motifs ($k_{nz} = 13$ for motif size is 3), the cost function decreases as the CEM^2 iterations proceed. When some components are pruned as described in the algorithm, the value of the cost function would increase to some extent. After some iterations, the remaining motifs will then be learned to better fit to the data, and thus the cost function decreases again. The number of motifs is automatically estimated by choosing with the one which gives the lowest cost function value. For synthetic data, the mixture model with 5 motifs gives the lowest cost function value, which is consistent to that estimated using the basic EM. In the four social networks we used, the cost functions have the lowest values when the motif numbers become 3, 3, 3 and 5 respectively, which are also consistent to the results observed in Section 3.2, but here we need to run CEM^2 only once. Fig. 5 shows the evolution of motifs annihilation by taking the Wiki-Vote network as an example. Figs. 5(a) - 5(e) are the motifs when the number of motifs is 5. With the iteration continues until convergence, the number of motifs becomes to 3, the corresponding motifs are listed in Figs. 5(f) - 5(h).



(a) Synthetic network (b) Amazon network (c) Wiki-Vote network

(d) Slashdot network (e) Epinions network

Fig. 4. The evolution of cost functions $L(\boldsymbol{\Theta}, \boldsymbol{X})$ until convergence in different datasets, the x-axis gives iteration times, k_{nz} means the number of none-zero components



(a) evo1motif1 (b) evo1motif2 (c) evo1motif3 (d) evo1motif4 (e) evo1motif5

(f) evo2motif1 (g) evo2motif2 (h) evo2motif3

Fig. 5. The evolution of motif pruning by taking Wiki-Vote network as an example

3.4 Computational Complexity

The overall complexity include those for subgraph sampling, generating of random networks, and the parameter estimation via the EM algorithms.

The complexity of sampling subgraphs of n nodes in a network is $R_S = O(N_s K^{n-1} n^{n+1})$, where K is a small constant value corresponding to the average node degree in the network, and N_s is the number of subgraphs sampled. The background model is simulated by randomized networks which generated by the switch method as in [2,11], where many rounds of "switchings" of two edges randomly selected from the real network are conducted while keeping the in/out degree of each node fixed. In so doing, the complexity of generating a random network is $O(T_s N_e)$ (the number of switches), where T_s is the switch times per edge (a random number in the range of $100-200$) and N_e is the number of edges in the real network. Overall time complexity for pre-processing is $O(N_s K^{n-1} n^{n+1}(1+N_r)+N_r T_s N_e)$, where N_r is the number of random networks.

For the basic EM algorithm, the complexity of each iteration is $O(n^2 N_s)$. So, the total complexity of EM algorithm together with pre-processing is $O(N_s \times K^{n-1} n^{n+1}(1+N_r)+N_r \times T_s \times N_e+k \times I \times n^2 N_s)$, where I is the iteration times of EM algorithm and k is optimal number of motifs. For CEM^2, it is only slightly computationally heavier than the basic EM algorithm due to the multiple E-steps to recompute $E[Z_h^w]$ [16]. As updating $E[Z_h^w]$ needs only full computation of Eq.(4) for $j = h$. For $j \neq h$, the terms $(X^w|\theta_h)$, which could contribute a lot to the computational cost of E-step, remain unchanged and thus only need to be computed once per sweep, like in the basic EM. However, the basic EM should be run several times with different motif numbers. CEM^2 is needed to run only once. So, the overall time complexity of CEM^2 is lighter than basic EM.

For further speedup, as the data are assumed independent and identically distributed ($i.i.d$) and thus can be partitioned into multiple subsets, our method can also take the advantage of parallel computing on GPUs [20] so as to be more scalable to large-scale datasets.

4 Conclusion and Future Works

Motif detection provides an important tool to assist the study of structural properties in network data for domains like bioinformatics and on-line social media. We proposed the use of the finite mixture model to detect multiple stochastic motifs in network data and a related CEM algorithm for automatically determining the optimal number of motifs embedded and the model parameters of the motifs. We applied the method to both synthetic and several benchmark datasets and discussed how the obtained motifs could be used to gain an in-depth understanding of the underlying stochastic local interaction patterns.

Our method works well for analyzing the network structural properties based on small motifs (i.e., 3 or 4 nodes). For future work, more scalable (possibly parallel) implementation will be needed if the analysis is to be carried out for motifs of various sizes. From the perspective of further improving modeling and thus the analysis power, related research directions include: 1) extending the method to take into consideration the sign of the edges, and 2) incorporating the timing information on edges to detect temporal motifs as a family of similar interaction patterns which over-represented throughout the period of time.

Acknowledgments. This work was supported by the General Research Fund (HKBU210410) from the Research Grant Council of the Hong Kong Special Administrative Region, China.

References

1. Shen-Orr, S., Milo, R., Mangan, S., Alon, U.: Network motifs in the transcriptional regulation network of Escherichia coli. Nature Genetics 31(1), 64–68 (2002)
2. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovski, D., Alon, U.: Network motifs: Simple building blocks of complex networks. Science 298(5594), 824–827 (2002)
3. Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., Ayzenshtat, I., Sheffer, M., Alon, U.: Superfamilies of evolved and designed networks. Science 303(5663), 1538–1541 (2004)
4. Mangan, S., Alon, U.: Structure and function of the feedforward loop network motif. PNAS USA 100(21), 11980–11985 (2003)
5. Berg, J., Michael, L.: Local graph alignment and motif search in biological networks. PNAS USA 101(41), 14689–14694 (2004)
6. Jiang, R., Tu, Z., Chen, T., Sun, F.: Network motif identification in stochastic networks. PNAS USA 103(25), 9404–9409 (2006)
7. Kashtan, N., Itzkovitz, S., Milo, R., Alon, U.: Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. Bioinformatics 20(11), 1746–1758 (2004)
8. Juszczyszyn, K., Kazienko, P., Musiał, K.: Local Topology of Social Network Based on Motif Analysis. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part II. LNCS (LNAI), vol. 5178, pp. 97–105. Springer, Heidelberg (2008)
9. Musial, K., Juszczyszyn, K.: Motif-based analysis of social position influence on interconnection patterns in complex social network. In: Proceedings of First Asian Conference on Intelligent Information and Database Systems, pp. 34–39 (2009)

10. Jiang, R., Chen, T., Sun, F.: Bayesian models and Gibbs sampling strategies for local graph alignment and motif identification in stochastic biological networks. Communications in Information & Systems 9(4), 347–370 (2009)
11. Liu, K., Cheung, W.K., Liu, J.: Stochastic network motif detection in social media. In: Proceedings of 2011 ICDM Workshop on Data Mining in Networks (2011)
12. McKay, B.: Nauty user's guide (version 2.4). Australian National University (2007)
13. Garey, M., Johnson, D.: Computers and intractability: A guide to the theory of np-completeness. Freeman San Francisco (1979)
14. Huan, J., Wang, W., Prins, J.: Efficient mining of frequent subgraphs in the presence of isomorphism. In: Proceedings of 2003 IEEE ICDM, pp. 549–552 (2003)
15. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. J. of the Royal Statistical Society. Series B 39(1), 1–38 (1977)
16. Figueiredo, M., Jain, A.: Unsupervised learning of finite mixture models. IEEE Transactions on PAMI 24(3), 381–396 (2002)
17. Wallace, C., Dowe, D.: Minimum message length and kolmogorov complexity. The Computer Journal 42(4), 270–283 (1999)
18. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Signed networks in social media. In: Proceedings of the 28th International Conference on Human Factors in Computing Systems, pp. 1361–1370 (2010)
19. Leskovec, J., Adamic, L., Huberman, B.: The dynamics of viral marketing. ACM Transactions on the Web 1(1), 5–44 (2007)
20. Kumar, N., Satoor, S., Buck, I.: Fast parallel expectation maximization for gaussian mixture models on gpus using cuda. In: 11th International Conference on High Performance Computing and Communications, pp. 103–109 (2009)

Scalable Similarity Matching in Streaming Time Series

Alice Marascu[1], Suleiman A. Khan[2], and Themis Palpanas[3]

[1] University of Trento
[2] Aalto University
[3] University of Trento
marascu@disi.unitn.eu, suleiman.khan@aalto.fi,
themis@disi.unitn.eu

Abstract. Nowadays online monitoring of data streams is essential in many real life applications, like sensor network monitoring, manufacturing process control, and video surveillance. One major problem in this area is the online identification of streaming sequences similar to a predefined set of pattern-sequences.

In this paper, we present a novel solution that extends the state of the art both in terms of effectiveness and efficiency. We propose the first online similarity matching algorithm based on Longest Common SubSequence that is specifically designed to operate in a streaming context, and that can effectively handle time scaling, as well as noisy data. In order to deal with high stream rates and multiple streams, we extend the algorithm to operate on multilevel approximations of the streaming data, therefore quickly pruning the search space. Finally, we incorporate in our approach error estimation mechanisms in order to reduce the number of false negatives.

We perform an extensive experimental evaluation using forty real datasets, diverse in nature and characteristics, and we also compare our approach to previous techniques. The experiments demonstrate the validity of our approach.

Keywords: data stream, online similarity matching, time series.

1 Introduction

In the last years, due to accelerated technology developments, more and more applications have the ability to process large amounts of streaming time series in real time, ranging from manufacturing process control and sensor network monitoring to financial trading [1] [2] [3] [4] [5]. A challenging task in processing streaming data is the discovery of predefined pattern-sequences that are contained in the current sliding window. This problem finds multiple applications in diverse domains, such as in network monitoring for network attack patterns, and in industrial engineering for faulty devices and equipment failure patterns. Previous work on streaming time series similarity [6] [7] proposed solutions that are limited either by the flexibility of the similarity measures, or by their scalability (these points are discussed in detail in Section 2).

Motivated by these observations, in this paper we propose a new approach that overcomes the above drawbacks. First, we observe that in the absence of a time scaling constraint, degenerate matches may be obtained (i.e., by matching points in

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 218–230, 2012.

the time series that are too far apart from each other). In order to address this problem, we introduce the notion of the *Continuous Warping Constraint* that specifies the maximum allowed time scaling, and thus, offers only meaningful results.

We propose the first adaptation of the Longest Common SubSequence (LCSS) similarity measure to the streaming context, since it has been shown that LCSS is more robust to noise (outliers) in time series matching [8].

In order to enable the processing of multiple streams, we introduce a framework based on Multilevel Summarization for the patterns and for the streaming time series. This technique offers the possibility to quickly discard parts of the time series that cannot lead to a match. Our work is the first to systematically study the required sliding window sizes of these multilevel approximations, as well as take into account and compensate for the errors introduced by these approximations. Therefore, we avoid false negatives in the final results.

Figure 1 is a schematic illustration of our approach. The streaming time series subsequences included in the current streaming window are summarized using a multilevel summarization method and the same operation is performed on the predefined pattern sequences. Then, the different levels of these summaries are compared using a streaming algorithm with very small memory footprint.

The main contributions of this paper can be summarized as follows.

- We propose the first method that employs the LCSS distance measure for the problem of time series similarity matching and is specifically designed to operate in a streaming context.
- We introduce the Continuous Warping Constraint (Sections 3.1.1and 3.1.2) for the online processing setting in order to overcome the unlimited time scaling problem.
- We describe a scalable framework for streaming similarity matching, based on streaming time series summarization. We couple these techniques with analytical results (Sections 3.1.3 and 3.2.2) and corresponding methods that compensate for the errors introduced by the approximations, ensuring high precision and recall with low runtimes.

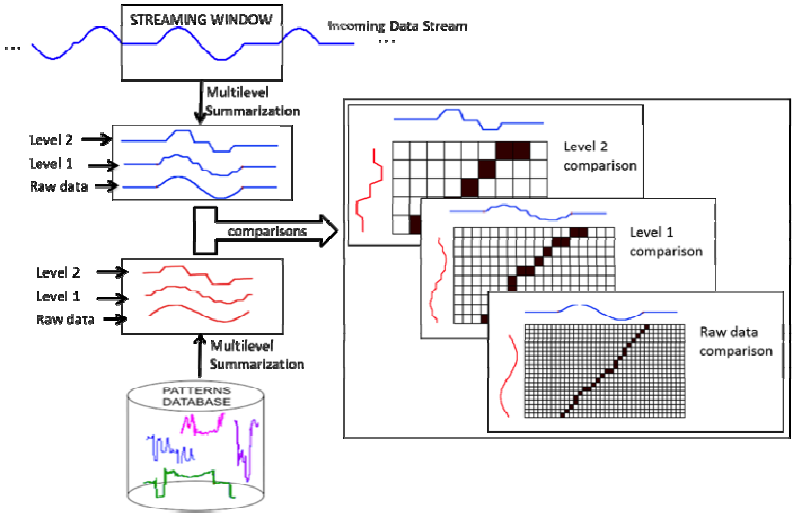


Fig. 1. Data stream monitoring for predefined patterns

- Finally, we perform an extensive experimental evaluation with forty real datasets (Section 4). The results demonstrate the validity of our approach, in terms of quality of results, and show that the proposed algorithms run (almost) three times faster than SPRING [9] (the current state of the art).

The rest of the paper is organized as follows. We start by briefly presenting the background and the related work in Section 2. Section 3 presents our approach and describes in detail our algorithm. Section 4 discusses the experimental evaluations, and Section 5 concludes the paper.

2 Background and Related Work

We now introduce some necessary notation, and discuss the related work.

A **time series** is an ordered sequence of n real-valued numbers $T=(t_1, t_2, ..., t_n)$. We consider t_1 the first element and t_n the last element of the time series. In the streaming case, new elements arrive continuously, so the size of the time series is infinite. In this work, we are focusing on a sliding window of the time series, containing the latest k streaming values. We also define a **subsequence** $t_{i,j}$ of a time series $T=(t_1, t_2, ..., t_n)$ is $t_{i,j}=(t_i, t_{i+1}, ..., t_j)$, such that $1 \le i \le j \le n$. We say that two subsequences are **similar** if the distance D between them is less than a user specified threshold ε.

2.1 Distance Measures

The most frequently used distance measure is the Euclidean distance, which computes the square root of the sum of the squared differences between all the corresponding elements of the two sequences (Figure 2(a)). The Euclidean distance cannot be applied on sequences of different sizes, or for element temporal shifts and, in these cases, the optimal alignment is achieved by DTW (Dynamic Time Warping) [10] (Figure 2(b)). DTW is an elastic distance that allows an element of one sequence to be mapped to multiple elements in the other sequence. If no temporal bound is applied, DTW can lead to pathological cases [11] [12] where very distant elements are allowed to be aligned. To avoid this problem, temporal constraints can be added in order to restrict the allowed temporal area for the alignment; the most used constraints are the Sakoe-Chiba band [13] and the Itakura Parallelogram band [14] (shaded area in Figure 3). The LCSS measure [8] is also an elastic distance measure that has an additional feature compared to DTW: it allows gaps in the alignment. This feature can be very valuable in real applications, since in this way we can model noise, outliers, and missing values (Figure 2(c)).

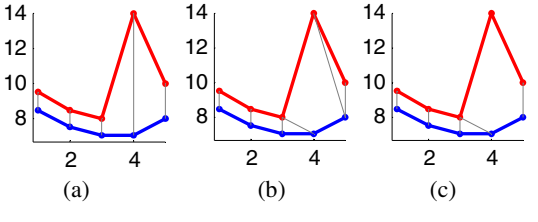


Fig. 2. Time series distances: Euclidean (a), DTW(b), and LCSS (c)

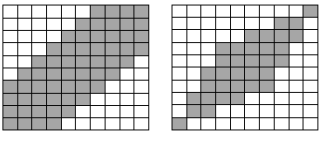


Fig. 3. Sakoe-Chiba band (left) and Itakura Parallelogram (right)

2.2 Similarity Matching

The Euclidean distance is used by [6] for identifying similar matches in streaming time series. This study proposes a technique, where the patterns to be matched are first hierarchically clustered based on their minimum bounding envelopes. Since this technique is based on the Euclidean distance, it requires the sequences to be of the same size and is rather sensitive to noise and distortion [15]. As a solution, [9] presented the SPRING algorithm that computes the DTW distance in a streaming data context. This algorithm allows comparison of sequences with different lengths and local time scaling, but does not efficiently handle noise (outlier points). We elaborate more on SPRING in the next subsection.

The warping distance techniques are also studied by [16], who propose the Spatial Assembling Distance (SpADe), a new distance measure for similarity matching in time series, which is able to operate incrementally in a streaming environment. However, Ding et al. [11] showed (based on a large collection of datasets) that DTW, in general, has better accuracy than SpADe.

Stream-DTW (SDTW) is proposed by [7] with the aim of having a fast DTW for streaming times series. SDTW is updatable with each new incoming data sequence. Nevertheless, the experimental results show that it is not faster than SPRING. In [17] the LCSS distance was used to process data streams, but was not adapted for online operation in a streaming context, which is the focus of our paper.

Other related works to this topic focused on approximations, thus proposing approximate distance formula. One example is the method proposed by [18] that uses the Boyer Moore string matching algorithm to match a sequence over a data stream. This approach is limited in that it operates with a single pattern and a single stream at a time. A multi-scale approximate representation of the patterns is proposed by [19] in order to speed up the processing. Even though the above representations introduce errors, neither these errors nor the accuracy of the proposed technique are explicitly studied.

2.3 SPRING Overview

The basic idea of SPRING (for more details see [9]) is to maintain a single, advanced form of the DTW matrix, called Sequence Time Warping Matrix (STWM). This matrix is used to compute the distances of all possible sequence comparisons simultaneously, such that the best matching sequences are monitored and finally reported when the matching is complete.

Each cell in the STWM matrix contains two values: the DTW distance $d(t,i)$ and the starting time $s(t,i)$ of sequence (t,i), where $t=1,2,\ldots n$ and $i=1,2,\ldots m$ are the time index in the matrix of the stream and of the pattern respectively. A subsequence starting at $s(t,i)$ and ending at the current time t has a cumulative DTW distance $d(t,i)$, and it is the best distance found so far after comparing the prefix of the stream sequence from time $s(t,i)$ to t, and the prefix of the pattern sequence from time 1 to i. On arrival of a new data point in the stream, the values of $d(t,i)$ and $s(t,i)$ are updated using Equations (1) and (2). A careful implementation of the SPRING algorithm leads to a space complexity of $O(m)$ and time complexity of $O(mn)$, just like DTW.

$$d(t,i) = \left\| a_t, b_i \right\| + d_{best}, \quad d(t,0) = 0, d(0,i) = \infty \tag{1}$$

$$d_{best} = \min \begin{cases} d(t-1,i-1) \\ d(t-1,i) \\ d(t,i-1) \end{cases} \quad \begin{array}{l} where: \\ t = 1,2,...,n \\ i = 1,2,...,m \end{array}$$

$$s(t,i) = \begin{cases} s(t-1,i-1) & if\ d(t-1,i-1) = d_{best} \\ s(t-1,i) & if\ d(t-1,i) = d_{best} \\ s(t,i-1) & if\ d(t,i-1) = d_{best} \end{cases} \tag{2}$$

3 Our Approach

In this section, we present two novel algorithms for streaming similarity matching called naiveSSM (naive Streaming Similarity Matching) and SSM (Streaming Similarity Matching).

The naiveSSM algorithm efficiently detects similar matches thanks to three features: it constrains the time scaling allowed in the matches, thus, avoiding degenerate answers; it handles outlier points in the data stream, by using LCSS; and it uses a special hierarchical summarization structure that allows it to effectively prune the search space. Although naiveSSM provides good results, it is not aware of the computation error introduced by the summarization method. SSM takes care of the computation error and improves the results by using a probabilistic error modeling feature.

3.1 The naiveSSM Algorithm

3.1.1 CWC Bands (Continuous Warping Constraint bands)

We observe that the simple addition of a Sakoe-Chiba band for solving the problem of degenerate matches would not be enough, since not all matching cases would be detected. Figure 4 illustrates this idea; two sequences situated outside of the band, but very near of its bounds, are not detected as matching because they are outside of the allowed area. The solution we propose is a novel formulation of Sakoe-Chiba band, which we call Continuous Warping Constraint (CWC band).

CWC band consists of multiple succeeding overlapping bands where each of these bands is bounding one possible matching sequence (Figure 5). More precisely, we propose to associate a boundary constraint to each possible matching, and not a general allowed area (as in Figure 4). In this way, the CWC band provides more flexible bounds that follow the matching sequences behavior. Figure 5 shows three CWC bands. The first matching sequence (dark grey) falls within the first CWC band, while the second sequence falls in the third CWC band, hence both being successfully detected as matching. A single CWC band is an envelope created around the pattern (the left allowed time scaling value being equal to the right allowed time scaling value); the size of the envelope is a user-defined parameter. The CWC bands have the additional advantage that they can be computed with negligible additional cost.

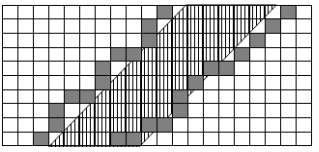


Fig. 4. Sakoe-Chiba band (shaded area), and two candidate matching sequences (dark grey) falling outside the constraint envelope

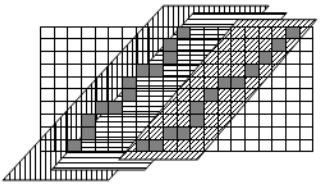


Fig. 5. The same two candidate matching sequences as Figure 4, and three overlapping CWC bands

The CWC bands can be added to the SPRING algorithm, on top of the DTW. Due to lack of space, in the following, we only discuss the application of CWC on top of streaming LCSS.

3.1.2 LCSS in a Streaming Context

LCSS provides a better support for noise compared to DTW, as we mention in Section 2. Equation (3) shows the LCSS computation of two sequences, A and B, of length n and m, respectively. The parameter γ is a user-defined threshold for the accepted distance. We now derive a novel formula for the streaming version of LCSS.

$$LCSS(A,B) = \begin{cases} 0 & if\ (A\ or\ B\ is\ Empty) \\ 1 + LCSS(a_{t-1}, b_{i-1}) & if\ (dist(a_t, b_i) < \gamma) \\ \max[LCSS(a_{t-1}, b_i), & otherwise \\ \quad LCSS(a_t, b_{i-1})] & \end{cases} \quad (3)$$

$$where \quad t = 1,2,...,n \quad and \quad i = 1,2,...,m$$

Since LCSS and DTW have similar matrix-based dynamic programming solutions (for the offline case), one may think that they also share the idea of the STWM matrix, thus, leading to a simple solution of replacing DTW with LCSS in the SPRING framework. Unfortunately, this is not a suitable solution: simply plugging LCSS Equation (3) into SPRING introduces false negatives and degenerate time scaled matches. The false negatives occur because the *otherwise* clause in Equation (3) selects the maximum of the two preceding sequences irrespective of the portion of the δ time scaling they have consumed. Therefore, it is possible that the selected sequence may have a higher LCSS value than the discarded sequence, but has already exceeded its allowed time scaling limit. The problem is that even though such a sequence will never become a matching sequence (because of its length), it may prevent a valid sequence from becoming a match.

To address this problem, we formulate the new CWC band constrained LCSS Equation (4) (due to lack of space, we omit the intermediate steps of deriving these new equations). In Equation (4), δ is a user defined parameter defining the maximum time scaling limit for the CWC bands, which corresponds to the size of the bands. The LCSS count is incremented only if the current pattern and stream value match, that is, they have a point to point distance less than the threshold γ, and the preceding diagonal LCSS falls within the CWC band (i.e., belongs to the allowed envelope area). Equation (5) describes the corresponding update of the starting time.

$$l(t,i)=\begin{cases}1+l(t-1,i-1) & if\begin{pmatrix} dist(a_t,b_i)<\gamma \\ \left|\left(t-s(t-1,i-1)+1\right)-i\right|\le\delta \end{pmatrix} \\ \max\begin{pmatrix} l(t-1,i)\times\left(\left|\left(t-s(t-1,i)+1\right)-i\right|\le\delta\right), \\ l(t,i-1)\times\left(\left|\left(t-s(t,i-1)+1\right)-i\right|\le\delta\right), \\ l(t-1,i-1)\times\left(\left|\left(t-s(t-1,i-1)+1\right)-i\right|\le\delta\right) \end{pmatrix} & otherwise \end{cases} \quad (4)$$

$$l(t,0)=0;\ l(0,i)=0 \quad where\ t=1,2,...,n;\ i=1,2,...,m$$

$$s(t,i)=\begin{cases} s(t-1,i-1) & if\left(l(t,i)=l(t-1,i-1)\right) \\ s(t-1,i) & if\left(l(t,i)=l(t-1,i)\right) \\ s(t,i-1) & if\left(l(t,i)=l(t,i-1)\right) \end{cases} \quad (5)$$

3.1.3 Multilevel Summarization

When trying to identify candidate matches of a given pattern in a streaming time series, we are bound to waste a significant amount of computations on testing subsequences that in fact cannot be a solution. In this subsection, we describe how we can effectively prune the search space by using a multilevel summarization structure on top of the streaming time series. Figure 6 illustrates the idea of a multilevel hierarchical summary (levels 1 and 2 in this figure), depicted in black-colored lines, of a streaming time series (level 0), depicted in black-colored points.

In this study, we use PAA[1] [20] as a summarization method, because of its simplicity, effectiveness, and efficiency in a streaming environment. However, other summarization methods can also be used. The algorithm operates in an incremental fashion. It processes the incoming values in batches and waits till the number of data points received is sufficient to build a complete new top level approximation segment, at which point the approximations of all levels are computed at once.

The benefit of employing this hierarchical summary structure is that the similarity matching can be executed on the summaries of the streaming time series, instead of the actual values, whose size is considerably larger. If the algorithm finds a candidate match at a high level of approximation, then it also checks the lower levels, and if necessary the actual stream as well.

Note that since we are using an elastic distance measure (i.e., LCSS), it is possible that the matching at lower approximation levels, and especially at the actual pattern or stream, may yield different starting and ending points. If not treated properly, this situation may lead to false negatives.

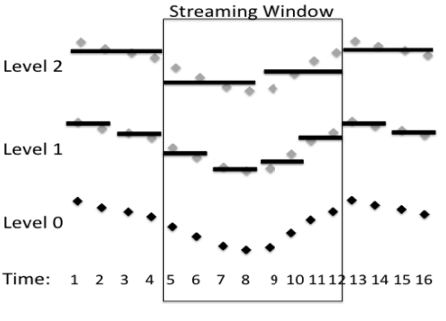


Fig. 6. Multilevel Summarization

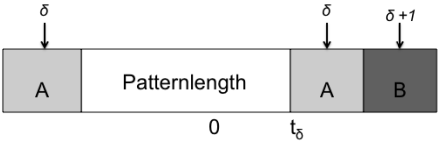


Fig. 7. Sequence Placement in Streaming Window at t_δ

[1] PAA (Piecewise Aggregate Approximation) divides the time series into N equal segments and approximates each segment by its average value.

First of all, the window size must be at least as big as the searched pattern. Then, since we use elastic matching, a candidate matching sequence can be extended up to δ data points in both directions (i.e., to the left and right), which gives the following inequality for the window size (refer to area "A" in Figure 7): $WindowSize \geq patternLength + 2\delta$. When a candidate matching sequence is detected, we must determine its locally optimal neighbour. For this purpose, and since δ is the maximum allowed time scaling, the window has to contain an extra δ data points (depicted as "B" in Figure 7). Finally, one more data point is needed for the optimality verification of the candidate sequence. The above statements lead to the following inequality for the window size: $WindowSize = patternLength + 3\,\delta + 1$.

3.2 SSM Algorithm

3.2.1 Probabilistic Error Modelling

As all approximations result in loss of information, multilevel summarization is also expected to lead to some loss of information. Therefore, it is highly probable that the set of candidate matches found at the highest level may *not* contain all the actual matches. One way of addressing this problem is by lower bounding the distance of the time series. Even though this is possible, this approach would lead to a computationally expensive solution. Instead, we propose an efficient solution based on the probability with which errors occur in our distance computations.

We randomly choose a sample of actual data point sequences from the streaming window. For all the sequences in the sample, we compute the error in the distance measure introduced by the approximation (by comparing to the distance computed based on the raw data). Then, we build a histogram that models the distribution of these errors, the Error Probability Distribution (EPD). Evidently, errors are smaller for the lower levels of approximation, since they contain more information, and are consequently more accurate than the approximations at higher levels.

The pruning decision of a candidate matching sequence is based on the EPD: we define the *error-margin* as 1-3σ (standard deviations) of the EPD. Intuitively, the error-margin indicates the difference that may exist between the distance computation based on the summarization levels and the true distance. Using an error-margin of 3σ, we have a very high probability that we are going to account for almost all errors.

Then, if the distance of a candidate sequence computed at one of the approximation levels is larger than the user-defined threshold ε, but less than $\varepsilon + error\text{-}margin$, this sequence remains a candidate match (and is further processed by the lower approximation levels).

3.2.2 Change-Based Error Monitoring

The data stream characteristics change continuously over time and EPD must reflect them. For this purpose, we set up a technique allowing the effective streaming computation of EPD. The most expensive part of the EPD computation is the computation of the LCSS distance between the pattern and all the samples. The continuous computation of EPD can be avoided by setting up a mechanism that will trigger the EPD re-computation only when the data distribution changes significantly.

In the work, we propose to use a technique that is based on the mean and variance of the data. This technique was proven to be effective [21] [22] and can be integrated in a streaming context. Though, more complex techniques for detecting data distribution changes can be applied as well [23].

Our algorithm operates as follows. When sufficient data arrives in the stream window, the EPD is constructed. Then the mean and the variance of the original streaming data are computed and registered together with the error margins. We consider that the EPD needs to be updated (i.e., reconstructed), only when the mean and variance of the current window changes by more than one standard deviation compared to the previous value. We will call this technique *change-based error monitoring*.

The only remaining question to answer is how often to sample. If we look for a pattern of size k within a streaming window of size n, there are $(n - k + 1)$ possible matching sequences for the first element of the window, $(n - k + 1 - 1)$ possible matching sequences for the second element of the window and so on until there is 1 single possible matching sequence for the n-k element of the window. Therefore there are $(n-k+1)+(n-k+1-1)+(n-k+1-2)+...+1=(n-k+1)*(n-k+2)/2$ possible matching sequences.

Given the large number of possible matching sequences, even a very small sampling rate (i.e., less than 1%) can be sufficient for the purpose of computing EPD. In the following section, we experimentally validate these choices.

4 Experimental Evaluation

All experiments were performed on a server configured with 4xGenuine Intel Xeon 3.0 GHz CPU, and 2 GB RAM, running the RedHat Enterprise ES operating system. The algorithm was coded in Matlab.

We used forty real datasets (for details, see [24]) with diverse characteristics from the UCR Time Series Repository [25], and treated them as streams. Patterns are randomly extracted from the streams, and the experiments are organized as follows. A dataset consists of several streams and several patterns. An experiment carried out on a single dataset means that each pattern in that dataset is compared with each stream in the same dataset. All experiments are carried out with patterns of length 50 (unless otherwise noted), and we report the averages over all runs, as well as the 95% Confidence Intervals (CIs).

We use precision and recall to measure accuracy: precision is defined as the ratio of true matches over all matches reported by the algorithm; recall is the ratio of true matches reported by the algorithm over all the true matches. The matches produced by SPRING with CWC bands serve as the baseline for all our experiments.

4.1 Approximate Similarity Matching

We first examine the performance of the naiveSSM algorithm. In this case, we use up to 5 levels for the multilevel summarization. The performance when using only some of these 5 levels of the summarization is lower (these results are omitted for brevity).

This is because level skipping results in more matches to be processed at lower levels, where processing is more expensive. We also did not observe any significant performance improvements when considering more than 5 levels.

Figure 8 shows the precision, recall and runtime of naiveSSM as a function of the pattern size, which ranges from 50 to 500 data points. We report results for naiveSSM for the cases where we have 3 (PAA3) and 5 (PAA5) approximation levels. We observe that naiveSSM scales linearly with the length of the patterns. The precision and recall results show that even though precision remains consistently high (averaging more than 98%), recall is rather low: naiveSSM using LCSS averages a recall of 90%.

These results confirm that the errors introduced by the approximation may lead to some matches being missed. Nevertheless, precision is high, because every candidate match is ultimately tested using the raw data as well. Finally, the results show that using 5 levels for the summarization leads to higher accuracy, but also higher running times (Figure 8(right)).

4.2 Compensating for Approximation Errors

We now present results for SSM, which aims to compensate for the errors introduced by the multilevel summarization, and thus, lead to higher recall values.

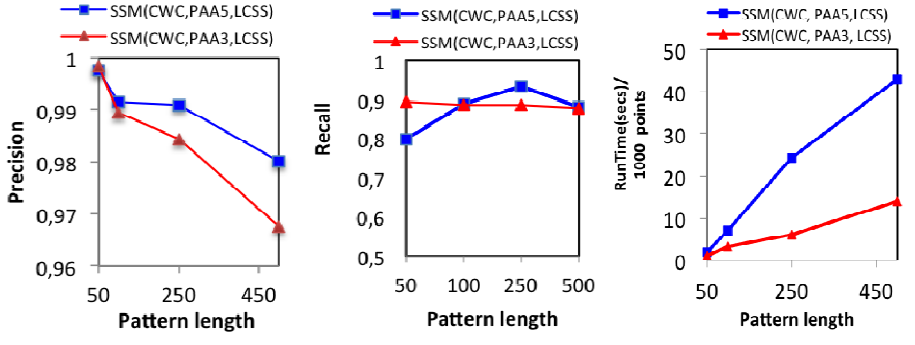


Fig. 8. Precision (left), Recall (middle), and Run Time (right) for different variants of naiveSSM

Table 1 shows the precision and recall numbers achieved by SSM with continuous and with change-based error monitoring, as a function of the error-margin. The results show that precision is in all cases consistently above 99%, while recall is over 95%, a significant improvement over naiveSSM. We also observe that SSM with change-based error monitoring performs very close to SSM with continuous monitoring (which is much more expensive). This validates our claims that the change-based error monitoring is an effective and efficient alternative to continuous error monitoring for producing high quality similarity matches. Finally, we observe that by increasing the error-margin from 1σ to 3σ, recall is only slightly improving. As expected, precision is unaffected.

We now study the role of the sampling rate on performance. Remember that this is the rate at which we sample the stream sequences during change-based error-monitoring, for computing the distance error introduced by the approximation and for building the EPD. In these experiments, we used an error-margin of 1σ, and sampling rates between 0.1% and 10%. Figure 9 presents the runtime of SSM as a function of the sampling rate. The results demonstrate that SSM with change-based error monitoring (curve with black squares) runs almost three times faster than the current state of the art (red, constant curve): SPRING (for fairness, with LCSS and CWC bands). Note that the time performance of SSM with continuous error monitoring (curve with dark blue circles) is better than only for very low sampling rates (0.1%).

It is also interesting to note that SSM performs very close to the lower bound represented by the naiveSSM algorithm (light grey curve), which does not use any error monitoring at all, and suffers in recall, averaging less than 90% (refer to Figure 8). In contrast, SSM achieves a significantly higher recall value, more than 95% (refer to Table 1). We note that the precision and recall of SSM with change-based monitoring remain stable, 99% and 96%, respectively, as the sampling rate varies between 0.1%-10% (results omitted for brevity). Overall, we can say that the SSM algorithm with change-based error monitoring is not only time efficient, but also highly accurate even when the error margin is small (1σ).

In the final experiment, we measure the number of distance computations performed by SSM with change-based error monitoring, using 3 levels of summarization. (Changing the sampling rate between 0.1%-10% does not affect the results.) We measured separately the number of computations for each level of summarization, including level 0: the raw data. The results show that the largest percentage of computations, 66%, occur at level 0 (18% at level 1, 12% at level 2, and 4% at level 3). These results signify that future attempts to further improve the runtime of the algorithm should focus on techniques for more aggressive, early (i.e., at higher levels) pruning of the candidate sequences.

Table 1. Precision and Recall for SSM with continuous and change-based error monitoring, as a function of the error margin.

	SSM continuous		**SSM change-based**	
error-margin	***Pr***	***R***	***Pr***	***R***
1	0.99	0.95	0.99	0.95
2	0.99	0.96	0.99	0.96
3	0.99	0.97	0.99	0.96

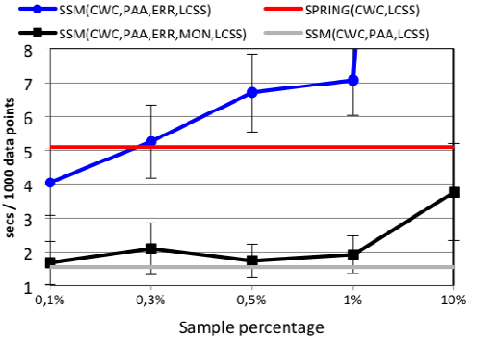


Fig. 9. Runtime vs Sample Percentage for SSM

5 Conclusions

In this work, we propose a new algorithm, able to efficiently detect similarity matching in a streaming context that is both scalable and noise-aware. Our

experiments on forty real datasets show that the proposed solution runs (almost) three times faster than previous approaches. At the same time, our solution exhibits high accuracy (precision and recall more than 99% and 95%, respectively), and ensures that we do not obtain degenerate answers, by employing the novel CWC bands.

Acknowledgements. This research was partially funded by FP7 EU IP project KAP (grant agreement no. 260111), and by Erasmus Mundus school EuMI (SAK).

References

1. Airoldi, E., Faloutsos, C.: Recovering latent time-series from their observed sums: network tomography with particle filters. In: KDD 2004 (2004)
2. Borgne, Y.-A.L., Santini, S., Bontempi, G.: Adaptive model selection for time series prediction in wireless sensor networks. Signal Process. 87(12), 3010–3020 (2007)
3. Zhu, Y., Shasha, D.: Statstream: statistical monitoring of thousands of data streams in real time. In: VLDB 2002 (2002)
4. Camerra, A., Palpanas, T., Shieh, J., Keogh, E.: iSAX 2.0: Indexing and Mining One Billion Time Series. In: ICDM 2010 (2010)
5. Dallachiesa, M., Nushi, B., Mirylenka, K., Palpanas, T.: Similarity Matching for Uncertain Time Series: Analytical and Experimental Comparison. In: QUeST 2011 (2011)
6. Wei, L., Keogh, E.J., Herle, H.V., Neto, A.M.: Atomic Wedgie: Efficient Query Filtering for Streaming Times Series. In: ICDM 2005, pp. 490–497 (2005)
7. Capitani, P., Ciaccia, P.: Warping the time on data streams. Data and Knowledge Engineering (62), 438–458 (2007)
8. Vlachos, M., Gunopulos, D., Kollios, G.: Discovering similar multidimensional trajectories. In: ICDE 2002, pp. 673–684 (2002)
9. Sakurai, Y., Faloutsos, C., Yamamuro, M.: Stream Monitoring under the Time Warping Distance. In: ICDE 2007 (2007)
10. Ratanamahatana, C.A., Keogh, E.: Everything you know about Dynamic Time Warping is Wrong. In: Third Workshop on Mining Temporal and Sequential Data 2004 (2004)
11. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures. In: VLDB 2008 (2008)
12. Salvador, S., Chan, P.: FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space. Intelligent Data Analysis 11(5), 561–580 (2007)
13. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. ASSP (1978)
14. Itakura, F.: Minimum Prediction Residual Principle Applied to Speech Recognition. ASSP 23, 52–72 (1975)
15. Agrawal, R., Faloutsos, C., Swami, A.N.: Efficient Similarity Search in Sequence Databases. In: Lomet, D.B. (ed.) FODO 1993. LNCS, vol. 730, pp. 69–84. Springer, Heidelberg (1993)
16. Chen, Y., Nascimento, M.A., Ooi, B.C., Tung, A.K.H.: SpADe: On Shape-based Pattern Detection in Streaming Time Series. In: ICDE 2007 (2007)
17. Marascu, A., Masseglia, F.: Mining Sequential Patterns from Data Streams: a Centroid Approach. J. Intell. Inf. Syst. 27(3), 291–307 (2006)
18. Harada, L.: Detection of complex temporal patterns over data streams. Information Systems 29(6), 439–459 (2004)

19. Lian, X., Chen, L., Yu, J.X., Wang, G., Yu, G.: Similarity Match Over High Speed Time-Series Streams. In: ICDE 2007 (2007)
20. Keogh, E.J., Chakrabarti, K., Pazzani, M.J., Mehrotra, S.: Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. Knowl. Inf. Syst. 3(3) (2001)
21. Babcock, B., Datar, M., Motwani, R.: Sampling From a Moving Window Over Streaming Data. In: SODA 2002 (2002)
22. Babcock, B., Datar, M., Motwani, R., O'Callaghan, L.: Maintaining Variance And k-medians Over Data Stream Windows. In: PODS, pp. 234–243 (2003)
23. Ben-David, S., Gehrke, J., Kifer, D.: Identifying Distribution Change in Data Streams. In: VLDB, Toronto, ON, Canada (2004)
24. Detailed list of datasets used, http://disi.unitn.eu/~themis/publications/pakdd12-ssm-appendix.pdf
25. UCR: Time Series Data Archive, http://www.cs.ucr.edu/~eamonn/time_series_data/

Scalable Mining of Frequent Tri-concepts from *Folksonomies*

Chiraz Trabelsi[1], Nader Jelassi[1], and Sadok Ben Yahia[1,2]

[1] Faculty of Sciences of Tunis, University Tunis El-Manar, 2092 Tunis, Tunisia
[2] Institut TELECOM, TELECOM SudParis, UMR 5157 CNRS Samovar, 91011 Evry Cedex, France
{chiraz.trabelsi,sadok.benyahia}@fst.rnu.tn

Abstract. Mining frequent tri-concepts from *folksonomies* is an interesting problem with broad applications. Most of the previous tri-concepts mining based algorithms avoided a straightforward handling of the triadic contexts and paid attention to an unfruitful projection of the induced search space into dyadic contexts. As a such projection is very computationally expensive since several tri-concepts are computed redundantly, scalable mining of *folksonomies* remains a challenging problem. In this paper, we introduce a new algorithm, called TRICONS, that directly tackles the triadic form of *folksonomies* towards a scalable extraction of tri-concepts. The main thrust of the introduced algorithm stands in the application of an appropriate closure operator that splits the search space into equivalence classes for the the localization of tri-minimal generators. These tri-minimal generators make the computation of the tri-concepts less arduous than do the pioneering approches of the literature. The experimental results show that the TRICONS enables the scalable frequent tri-concepts mining over two real-life *folksonomies*.

Keywords: Folksonomies, Triadic Concept Analysis, Closure Operator, Equivalence Classes, Triadic Concepts.

1 Introduction and Motivations

Complementing the Semantic Web effort, a new breed of so-called Web 2.0 applications recently emerged on the Web. Indeed, social bookmarking systems, such as *e.g.,* DEL.ICIO.US[1], BIBSONOMY[2] or FLICKR[3] have become the predominant form of content categorization of the Web 2.0 age. The main thrust of these Web 2.0 systems is their easy use that relies on simple, straightforward structures by allowing their users to label diverse resources with freely chosen keywords *aka* tags. The resulting structures are called *folksonomies*[4], that is, "taxonomies" created by the "folks". Considered as a tripartite hyper-graph [9] of tags, users and resources, the new data of *folksonomy* systems

[1] http://www.delicious.com
[2] http://www.bibsonomy.org
[3] http://www.flickr.com
[4] http://wwww.vanderwal.net/folksonomy.html

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 231–242, 2012.

provides a rich resource for data analysis, information retrieval, and knowledge discovery applications. Recently, the discovery of shared conceptualizations opens a new research field which may prove interesting also outside the *folksonomy* domain: closed tri-sets (triadic concepts) mining in triadic data [6]. Actually, this line of Triadic Concept Analysis did not grasp a broad attention. However, with the rise of *folksonomies*, formally represented as triadic contexts, many researches advocate the extraction of lossless concise representations of interesting patterns from triadic data.

In this paper, we are mainly interested in the mining of frequent triadic concepts (tri-concepts for short) from 3-dimensional data, *i.e., folksonomy*. These patterns are among the recent research topics in Triadic Concept Analysis. In this respect, a determined algorithmic effort was furnished to get out this type of patterns. Worth of mention, the pioneering work of Stumme *et al.,* through the TRIAS algorithm [6], for tri-concepts mining. TRIAS inputs a *folksonomy*, formally represented as a triadic context, and computes all tri-concepts. However, the main moan that can be addressed to TRIAS, stands in its need to transform the triadic context into dyadic contexts in order to extract tri-concepts. Thus, the mining task becomes very computationally expensive and could be avoided by extending the basic notions of *FCA* (Formal Concept Analysis) for the triadic case. Ji et *al.,* in [7], have proposed un alternative algorithm, called CUBEMINER, which directly operates on the triadic context. It consists in using cubes called *cutters* generalizing the cutters introduced for constraint-based mining of formal concepts [1]. Yet, in a *folksonomy*, the number of cutters may be very large as far as the cardinality of at least one dimension of a *folksonomy* is high. Besides, the CUBEMINER algorithm operates in a depth-first manner, which has the risk of causing infinite trees. More recently, Cerf *et al.*, in [2], proposed the DATA-PEELER algorithm with the challenge of beating both later algorithms in terms of performance. The DATA-PEELER algorithm is able to extract all closed concepts from n-ary relations. DATA-PEELER enumerates all the n-dimensional closed patterns in a depth first manner using a binary tree enumeration strategy. However, similarly to CUBEMINER, the strategy of DATA-PEELER, involving a depth-first approach implies its depth's recursion, in the worst case, to the total number of elements (whatever the dimension). Moreover, DATA-PEELER is hampered by the large number of elements that may contain any of the *folksonomy's* dimensions and its strategy becomes ineffective and leads to a complex computation of tri-concepts.

In this respect, a compelling and thriving issue is to introduce a new scalable algorithm, that overcomes the flaws of the previous ones. Hence, in this work, the main contribution is to introduce a new algorithm for tri-concepts mining, called TRICONS, aiming at providing better scalabilty than do the pioneering approches of the literature, by applying an appropriate closure operator. In fact, the closure operator splits the search space into equivalence classes in order to find the tri-minimal generators. These tri-minimal generators, representative of the different equivalence classes, make the computation of the tri-concepts less arduous than do the aforementioned ones. Indeed, the tri-minimal generators are the smallest elements, *i.e.,* tri-sets, in an equivalence class, while their associated closure is the largest one within the corresponding equivalence class. Thus, the pairs - composed by Tri-MGs and their related closures - allow, *(i)* an easier localization (extraction) of each tri-concept since it is necessarily

encompassed by an Tri-MG and the related closures and; *(ii)* to straightforwardly handle the triadic form of a *folksonomy* towards an efficient extraction of tri-concepts.

The remainder of the paper is organized as follows. Section 2 recalls the key notions used throughout this paper. We scrutinize the related work of mining triadic concepts in section 3. In section 4, we introduce a new closure operator to the triadic context as well as the TRICONS algorithm dedicated to the extraction of frequent tri-concepts. The empirical evidences about the performance of our approach are provided in Section 5. Finally, we conclude the paper with a summary and we sketch ongoing research in section 6.

2 Key Notions

In this section, we briefly sketch the key notions that will be of use in the remainder of this paper. In the following, we start by presenting a formal definition of a *folksonomy* [6].

Definition 1. (FOLKSONOMY) *A folksonomy is a set of tuples* $\mathcal{F} = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{Y})$, *where* $\mathcal{Y} \subseteq \mathcal{U} \times \mathcal{T} \times \mathcal{R}$ *is a triadic relation such as each* $y \subseteq \mathcal{Y}$ *can be represented by a triple:* $y = \{(u, t, r) \mid u \in \mathcal{U}, t \in \mathcal{T}, r \in \mathcal{R}\}$, *denoting that the user* u *annotated the resource* r *using the tag* t.

Example 1. An exemple of a *folksonomy* $\mathcal{F}$ is depicted by Table 1 with $\mathcal{U} = \{u_1, u_2, \ldots, u_7\}$, $\mathcal{T} = \{t_1, t_2, \ldots, t_5\}$ and $\mathcal{R} = \{r_1, r_2, r_3\}$. Each cross within the ternary relation indicates a tagging operation by a user from $\mathcal{U}$, a tag from $\mathcal{T}$ and a resource from $\mathcal{R}$, *i.e.*, a user has tagged a particular resource with a particular tag. For example, the user u_1 has assigned the tags t_2, t_3 and t_4, respectively, to the resources r_1, r_2 and r_3.

Table 1. A toy example of a *folksonomy* that would be of use throughout the paper

$\mathcal{U}/\mathcal{R}$-$\mathcal{T}$	r_1					r_2					r_3				
	t_1	t_2	t_3	t_4	t_5	t_1	t_2	t_3	t_4	t_5	t_1	t_2	t_3	t_4	t_5
u_1		×	×	×			×	×	×			×	×	×	
u_2		×	×	×		×	×	×	×		×	×	×	×	
u_3		×	×	×		×	×	×	×		×	×	×	×	
u_4						×			×		×			×	
u_5		×	×	×	×		×	×	×	×		×	×	×	
u_6				×	×				×	×					
u_7	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×

The following definition presents the frequent tri-set [6].

Definition 2. (A (FREQUENT) TRI-SET) *Let* $\mathcal{F} = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{Y})$ *be a folksonomy. A tri-set of* $\mathcal{F}$ *is a triple* (A, B, C) *with* $A \subseteq \mathcal{U}$, $B \subseteq \mathcal{T}$, $C \subseteq \mathcal{R}$ *such that* $A \times B \times C \subseteq \mathcal{Y}$. *A tri-set* (A, B, C) *of* $\mathcal{F}$ *is said* frequent *whenever* $|A| \geq minsupp_u$, $|B| \geq minsupp_t$ *and* $|C| \geq minsupp_r$, *where* $minsupp_u$, $minsupp_t$ *and* $minsupp_r$ *are user-defined thresholds.*

As the set of all frequent tri-sets is highly redundant, we will in particular consider a specific condensed representation, *i.e.,* a subset which contains the same information, namely the set of all frequent tri-concepts. The latter's definition is given in the following [6,8].

Definition 3. ((FREQUENT) TRIADIC CONCEPT) *A triadic concept (or a tri-concept for short) of a* folksonomy $\mathcal{F} = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{Y})$ *is a triple* (U, T, R) *with* $U \subseteq \mathcal{U}$, $T \subseteq \mathcal{T}$, *and* $R \subseteq \mathcal{R}$ *with* $U \times T \times R \subseteq \mathcal{Y}$ *such that the triple* (U, T, R) *is maximal, i.e., for* $U_1 \subseteq U$, $T_1 \subseteq T$ *and* $R_1 \subseteq R$ *with* $U_1 \times T_1 \times R_1 \subseteq \mathcal{Y}$, *the containments* $U \subseteq U_1$, $T \subseteq T_1$, *and* $R \subseteq R_1$ *always imply* $(U, T, R) = (U_1, T_1, R_1)$. *A tri-concept is said to be frequent whenever it is a frequent tri-set. The set of all frequent tri-concepts of* $\mathcal{F}$ *is equal to* $\mathcal{TC} = \{TC \mid TC = (U, T, R) \in \mathcal{Y}$ *is a tri-concept*$\}$.

Given a tri-concept $\mathcal{TC} = (U, T, R)$, the U, R and T parts are respectively called ***Extent***, ***Intent***, and **Modus**.

Example 2. Consider the *folksonomy* depicted by table 1. We can denote that the tri-set S_1=$\{\{u_5, u_7\}, \{t_2, t_3, t_4\}, \{r_1, r_2\}\}$ is not a tri-concept of $\mathcal{F}$. Whereas, TC_1=$\{\{u_5, u_7\}, \{t_2, t_3, t_4\}, \{r_1, r_2, r_3\}\}$ is a tri-concept of $\mathcal{F}$: it includes all maximal tags and resources shared by the users u_5 and u_7.

3 Related Work

With the rise of *folksonomies*, formally represented as triadic contexts, many researches advocate the extraction of implicit shared conceptualizations formally sketched by tri-concepts. Indeed, Jäschke et *al.*, in [6], introduced the TRIAS algorithm to compute frequent tri-concepts from a *folksonomy*. Hence, tackling a *folksonomy* $\mathcal{F} = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{Y})$, TRIAS first constructs a dyadic context $\mathcal{K}_1 = (\mathcal{U}, \mathcal{T} \times \mathcal{R}, \mathcal{Y}_1)$ whose columns correspond to couples of elements from $\mathcal{T}$ and $\mathcal{R}$ and then, via a projection, according to the $\mathcal{T}$ and $\mathcal{R}$ axis, extracts formal concepts. The second step of TRIAS consists, for each formal concept, in checking whether it is closed *w.r.t.* $\mathcal{U}$. Actually, the main feature of TRIAS is to exploit the subsets of tri-concepts already extracted in order to check whether they lead to new tri-concepts. However, several tri-concepts are computed redundantly inducing a number of unnecessary computations. This drawback occurs because of the particular order of extraction of tri-concepts which is strongly inspired by the way of doing of the NEXTCLOSURE algorithm [4], dedicated to building of a lattice of frequent closed itemsets. Nevertheless, Ji et *al.,* in [7], have introduced an alternative algorithm called CUBEMINER, which directly operates on the triadic context. It consists in using cubes called *cutters* generalizing the cutters introduced for constraint-based mining of formal concepts in [1]. These cutters are recursively processed to generate candidates at each level, thus, the number of levels of the execution equals that of cutters. For each cutter applied to a tri-set, three candidates are constructed accordingly to the three axis of the *folksonomy* as long as the tri-set contains all elements of the current cutter. When no more cutter is applicable on a tri-set, it becomes a tri-concept. Yet, in a *folksonomy*, the number of cutters may be very large as far as the cardinality of at least one set of $\mathcal{F}$ is high. Besides, the CUBEMINER algorithm operates in a depth-first manner, which has the risk of causing infinite trees. Moreover, at each level, several checks

are performed on each candidate to ensure its closeness and its uniqueness which is very computationally expensive. Indeed, each candidate must be compared twice to the elements of the cutters. More recently, Cerf *et al.*, in [2], proposed the DATA-PEELER algorithm with the challenge of outperforming both TRIAS and CUBEMINER algorithms in terms of performance. The DATA-PEELER algorithm is able to extract closed concepts from n-ary relations by enumerating all the n-dimensional closed patterns in a depth first manner using a binary tree enumeration strategy. At each level, the current node of the tree is split into two nodes after selecting the element to be enumerated. In addition, the DATA-PEELER algorithm does not store the previously computed patterns in main memory for duplicate detection and closure checking. However, similarly to CUBEMINER, the strategy of DATA-PEELER, involving a depth-first approach, may cause infinite trees. Aiming at palliating these hindrances in effectively extracting tri-concepts, we introduce the TRICONS algorithm dedicated to an efficient extraction of frequent triadic concepts from a *folksonomy*. Following the minimum description length principle, the set of frequent tri-concepts represents a concise representation of frequent tri-sets, by providing the shortest description of the whole set of these frequent patterns. The main thrust of the TRICONS algorithm stands in the localisation of the smallest elements, *i.e.,* tri-sets, called *tri-Minimal generators* (Tri-MGs), in an equivalence class. Indeed, these Tri-MGs are the first reachable elements of their respective equivalence classes, thanks to a breadth-first sweeping of the associated search space. Doing so, makes the computation of the tri-concepts less arduous than do the aforementioned ones.

4 The TRICONS Algorithm

In this section, we firstly, introduce a new closure operator for a triadic context as well as an extension of the notion of minimal generator. Thereafter, we describe the TRICONS algorithm.

4.1 Main Notions of the TRICONS Algorithm

Lehmann and Wille have introduced in [8] two closure operators for the construction of triadic concepts. However, these operators are only of use on dyadic contexts, *i.e.*, the *folksonomy* should be split into three dyadic contexts. Hence, we introduce, in what follows, a new closure operator for a triadic context.

Definition 4. *Let $S = (A, B, C)$ be a tri-set of $\mathcal{F}$. A mapping h is defined as follows :*

$$h(S) = h(A, B, C) = (U, T, R) \mid U = \{u_i \in \mathcal{U} \mid (u_i, t_i, r_i) \in \mathcal{Y} \forall\, t_i \in B, \forall\, r_i \in C\}$$
$$\wedge\, T = \{t_i \in \mathcal{T} \mid (u_i, t_i, r_i) \in \mathcal{Y} \forall\, u_i \in U, \forall\, r_i \in C\}$$
$$\wedge\, R = \{r_i \in \mathcal{R} \mid (u_i, t_i, r_i) \in \mathcal{Y} \forall\, u_i \in U, \forall\, t_i \in T\}$$

Roughly speaking, $h(S)$ computes the largest tri-set in the *folksonomy* which contains maximal sets of tags and resources shared by a group of users containing A. For example, considering the *folksonomy* $\mathcal{F}$ depicted by Table 1, we have $h\{u_1, \{t_2, t_3, t_4\}, r_1\} = \{\{u_1, u_2, u_3, u_5, u_7\}, \{t_2, t_3, t_4\}, \{r_1, r_2, r_3\}\}$.

Proposition 1. *h is a closure operator.*

Proof. To prove that *h* is a closure operator, we have to prove that this closure operator fulfills the three properties of **extensivity**, **idempotency** and **isotony** [3].

(1) ***Extensivity***
Let $T = (A, B, C)$ be a tri-set of $\mathcal{F} \Rightarrow h(T) = (U, T, R)$ such that :
$U = \{u_i \in \mathcal{U} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ t_i \in B, \forall\ r_i \in C\} \supseteq A$ since we have $(u_i, t_i, r_i) \in \mathcal{Y}$ $\forall\ u_i \in A, \forall\ t_i \in B, \forall\ r_i \in C$,

$T = \{t_i \in \mathcal{T} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ r_i \in C\} \supseteq B$ since $U \supseteq A$
and $R = \{r_i \in \mathcal{R} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ t_i \in T\} \supseteq C$ since $U \supseteq A$ and $T \supseteq B$.
Then, $(A, B, C) \subseteq (U, T, R) \Rightarrow T \subseteq h(T)$

(2) ***Idempotency***
Let $T = (A, B, C)$ be a tri-set of $\mathcal{F} \Rightarrow h(T) = (U, T, R) \Rightarrow h(U, T, R) = (U', T', R')$ such that :

$U' = \{u_i \in \mathcal{U} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ t_i \in T, \forall\ r_i \in R\} = U$,
$T' = \{t_i \in \mathcal{T} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ r_i \in C\} = T$,
and $R' = \{r_i \in \mathcal{R} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ t_i \in T'\} = R$.
Then, $(U', T', R') = (U, T, R) \Rightarrow h(h(T)) = h(T)$

(3) ***Isotony***
Let $T = (A, B, C)$ and $T' = (A', B', C')$ be tri-sets of $\mathcal{F}$ with $T \subseteq T' \Rightarrow h(T) = (U, T, R)$ and $h(T') = (U', T', R')$ such that :

On the one hand, $U' = \{u_i \in \mathcal{U} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ t_i \in B', \forall\ r_i \in C'\}$.
and $U = \{u_i \in \mathcal{U} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ t_i \in B, \forall\ r_i \in C\}$.
$\Rightarrow U' \supseteq U$ since $B \subseteq B'$ and $C \subseteq C'$ [8].
On the other hand, $T = \{t_i \in \mathcal{T} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ r_i \in C\}$, $R = \{r_i \in \mathcal{R} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ t_i \in T\}$, $T' = \{t_i \in \mathcal{T} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ r_i \in C\}$ and $R' = \{r_i \in \mathcal{R} \mid (u_i, t_i, r_i) \in \mathcal{Y}\ \forall\ u_i \in U, \forall\ t_i \in T'\}$
$\Rightarrow T \subseteq T'$ since $U \subseteq U'$ and $R \subseteq R'$ since $U \subseteq U'$ and $T \subseteq T'$ [8].
Then, $(U, T, R) \subseteq (U', T', R') \Rightarrow h(T) \subseteq h(T')$

According to (1), (2) and (3), *h* is a closure operator.

Like the dyadic case [10], the closure operator induces an equivalence relation on the power set of elements, *i.e.,* tri-sets in the *folksonomy*, portioning it into disjoint subsets called *equivalence classes* that we introduce in the following :

Definition 5. (EQUIVALENCE CLASS) *Let $S_1 = (A_1, B_1, C_1)$, $S_2 = (A_2, B_2, C_2)$ be two tri-sets of $\mathcal{F}$ and $TC \in \mathcal{TC}$. S_1 and S_2 belong to the same equivalence class represented by the tri-concept TC, i.e., $S_1 \equiv_{TC} S_2$ iff $h(S_1) = h(S_2) = TC$.*

The smallest tri-set (*w.r.t.* the number of items) in each equivalence class is called a tri-minimal generator and is defined as follows:

Definition 6. (TRI-MINIMAL GENERATOR) *Let $g = (A, B, C)$ be a tri-set such as $A \subseteq \mathcal{U}$, $B \subseteq \mathcal{T}$ and $C \subseteq \mathcal{R}$ and $TC \in \mathcal{TC}$. The triple g is a tri-minimal generator (tri-generator for short) of TC iff $h(g) = TC$ and $\nexists\, g_1 = (A_1, B_1, C_1)$ such as :*

1. $A = A_1$,
2. $(B_1 \subseteq B \wedge C_1 \subset C) \vee (B_1 \subset B \wedge C_1 \subseteq C)$, *and*
3. $h(g) = h(g_1) = TC$.

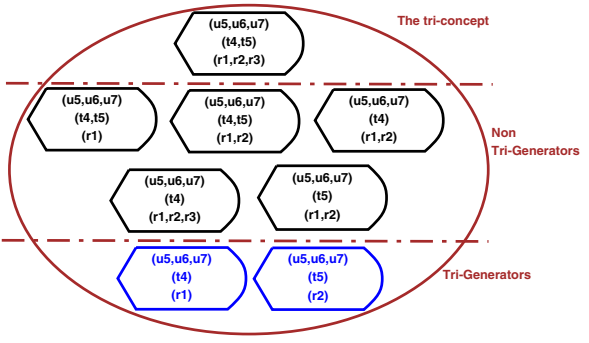


Fig. 1. Example of an equivalence class from $\mathcal{F}$

Figure 1 sketches a sample class of the induced equivalence relation from the *folksonomy* depicted by table 1. The largest unsubsumed tri-set $TC = \{\{u_5, u_6, u_7\}, \{t_4, t_5\}, \{r_1, r_2, r_3\}\}$, has three tri-generators g_1, g_2 and g_3. However, $g_4 = \{\{u_5, u_6, u_7\}, \{t_4, t_5\}, r_1\}$ is not a tri-generator of TC since it exists g_1 such as $g_1.extent$=$g_4.extent$, ($g_1.intent = \subseteq g_4.intent \wedge g_1.modus \subset g_4.modus$).

4.2 Description of the TRICONS Algorithm

TRICONS operates in three steps as follows:

1. The extraction of tri-generators;
2. The computation of the `modus` part of tri-concepts;
3. The computation of the `intent` part of tri-concepts.

The pseudo code of the TRICONS algorithm is sketched by Algorithm 1. TRICONS takes as input a *folksonomy* $\mathcal{F} = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{Y})$ as well as three user-defined thresholds : $minsupp_u$, $minsupp_t$ and $minsupp_r$. The TRICONS algorithm outputs the set of all frequent tri-concepts that fulfill these aforementioned thresholds. TRICONS operates as follows : it starts by invoking the **TRISORT** procedure (Line 2), that sorts the *folksonomy w.r.t.* the fields r, t and u, respectively. This sorting facilitates the handling of the *folksonomy* in order to extract the tri-generators. Then, TRICONS calls the FINDMINIMALGENERATORS procedure (Step 1), which pseudo-code is given by Algorithm 2, in order to extract the tri-generators which are stored in the set $\mathcal{MG}$ (Line 4) : for each triple (u, t, r), FINDMINIMALGENERATORS computes the set U_s which is the maximal set of users (including u) sharing the tag t and the resource r (Algorithm 2, Line 4).

ALGORITHM 1: TRICONS

Data :
1. $\mathcal{F}$: ($\mathcal{U}$, $\mathcal{T}$, $\mathcal{R}$, $\mathcal{Y}$) : A *Folksonomy*.
2. $minsupp_u$, $minsupp_t$, $minsupp_r$: User-defined thresholds.

Result : $\mathcal{TC}$: {Frequent tri-concepts}.

begin
  TRISORT($\mathcal{F}$);
  /*Step 1 : The extraction of tri-generators*/
  FINDMINIMALGENERATORS($\mathcal{F}$, $\mathcal{MG}$, $minsupp_u$);
  /*Step 2 : The computation of the modus part*/
  foreach *tri-gen* $g \in \mathcal{MG}$ **do**
    $Increase\_Set(\mathcal{MG}, minsupp_u, minsupp_t, g, \mathcal{TS}, true)$;
  end
  PRUNEINFREQUENTSETS($\mathcal{TS}$,$minsupp_t$);
  /*Step 3 : The computation of the intent part*/
  foreach *tri-set* $s \in \mathcal{TS}$ **do**
    $Increase\_Set(\mathcal{TS}, minsupp_u, minsupp_t, s, \mathcal{TC}, false)$;
  end
  PRUNEINFREQUENTSETS($\mathcal{TC}$,$minsupp_r$);
end
return $\mathcal{TC}$;

ALGORITHM 2: FINDMINIMALGENERATORS

Data :
1. $\mathcal{MG}$: The set of frequent tri-generators;
2. $\mathcal{F}$ ($\mathcal{U}$, $\mathcal{T}$, $\mathcal{R}$, $\mathcal{Y}$) : A *folksonomy*;
3. $minsupp_u$: User-defined threshold of user's support.

Result : $\mathcal{MG}$: {The set of frequent tri-generators}.

begin
  while *(u, t, r)* $\neq NULL$ **do**
    (u, t, r):=NEXTTRIPLE($\mathcal{F}$);
    U_s= $\{u_i \in \mathcal{U} \mid (u_i, t, r) \in \mathcal{Y}\}$;
    if $\mid U_s \mid \geq minsupp_u$ **then**
      $g.extent = U_s$; $g.intent = r$; $g.modus = t$;
      AddTri($\mathcal{MG}$, g)
    end
  end
end
return $\mathcal{MG}$;

Algorithm 2 invokes both **AddTri** and **NextTriple** functions. The first one allows to add the tri-set Tri to the set $\mathcal{S}$, whereas the second one returns for each call the next triple (u, t, r) of the *folksonomy* $\mathcal{F}$.

ALGORITHM 3: *Increase_Set*

Data :
1. $\mathcal{S}_{IN}$: The set of frequent tri-generators/tri-sets.
2. min_u, min_t : User-defined thresholds of extent and modus support.
3. tri : A tri-generator/tri-set.
4. *flag* : a boolean indicator.

Result : $\mathcal{S}_{OUT}$: {The set of frequent tri-sets/tri-concepts}.

1 **begin**
2 **foreach** *tri-set* $tri' \in \mathcal{S}_{IN}$ **do**
3 **if** flag ***and*** $tri.intent = tri'.intent$ ***and*** $tri.extent \subseteq tri'.extent$ **then**
4 $s.intent = g.intent$; $s.extent = g.extent$; $s.modus = g.modus \cup g'.modus$; AddTri($\mathcal{S}_{OUT}$, s);
5 **end**
6 **else if** flag ***and*** $tri.intent = tri'.intent$ ***and*** tri *and* tri' *are* incomparables **then**
7 $g''.extent = g.extent \cap g'.extent$; $g''.modus = g.modus \cup g'.modus$; $g''.intent = g.intent$; **If** $\mid g''.extent \mid \geq min_u$ **then** AddTri($\mathcal{MG}$, g'');
8 **end**
9 **else if** not flag ***and*** $tri.extent \subseteq tri'.extent$ ***and*** $tri.modus \subseteq tri'.modus$ ***and*** $tri.intent \neq tri'.intent$ **then**
10 $TC.extent = s.extent$; $TC.modus = s.modus$; $TC.intent = s.intent \cup s'.intent$; AddTri($\mathcal{S}_{OUT}$, TC);
11 **end**
12 **else if** not flag ***and*** tri *and* tri' *are* incomparables **then**
13 $s''.extent = s.extent \cap s'.extent$; $s''.modus = s.modus \cap s'.modus$; $s''.intent = s.intent \cup s'.intent$;
14 **If** $\mid s''.extent \mid \geq min_u$ **and** $\mid s''.modus \mid \geq min_t$ **then** AddTri($\mathcal{TS}$, s'');
15 **end**
16 **end**
17 **end**
18 **return** $\mathcal{S}_{OUT}$;

Afterwards, TriCons invokes the $Increase\_Set$ procedure (Step 2) for each tri-generator of $\mathcal{MG}$ (Lines 6-8), which pseudo-code is given by Algorithm 3, in order to compute the modus part of the tri-concepts. The two first cases of Algorithm 3 (Lines 3 and 6) have to be considered by $Increase\_Set$ according to the extent of each tri-generator before returning the set $\mathcal{TS}$ of tri-sets. The boolean indicator *flag* marked by TriCons shows whether the tri-set processed by the $Increase\_Set$ procedure is a tri-generator. Then, infrequent tri-sets, *i.e.*, whose the modus part cardinality does not fulfill the minimum threshold $minsupp_t$ are pruned (Line 9). In the third and final step, TriCons invokes a second time the $Increase\_Set$ procedure for each tri-set of $\mathcal{TS}$ (Lines 11-13), in order to compute the intent part. $Increase\_Set$ looks for tri-sets s' of $\mathcal{TS}$ having a different intent part than a given tri-set s (Algorithm 3, Line 9). Before

returning the set $\mathcal{TC}$ of tri-concepts, TRICONS prunes the infrequent ones, *i.e.*, whose the intent cardinality does not fulfill the minimum threshold $minsupp_r$ by invoking the **PRUNEINFREQUENTSETS** procedure (Line 14). TRICONS comes to an end after invoking this procedure and returns the set of the frequent tri-concepts which fulfills the three thresholds $minsupp_u$, $minsupp_t$ and $minsupp_r$.

Example 3. Considering the *folksonomy* depicted by Table 1 (page 4) with $minsupp_u$ = 3, $minsupp_t$ = 3 and $minsupp_r$ = 2 yields the following track for the TRICONS algorithm. The first step of TRICONS consists in the extraction of (frequent) tri-generators from the context (step 1) thanks to the FINDMINIMALGENERATORS procedure. Then, invoking firstly the $Increase\_Set$ procedure, on these tri-generators, allows the reduction of the number of candidates. Hence, only five candidates, at step 2, are generated which directly lead to the frequent tri-concepts extracted by TRICONS. So, the set $\mathcal{TS}$ contains the tri-sets $\{\{u_1, u_2, u_3, u_5, u_7\}, \{t_2, t_3, t_4\}, r_1\}$, $\{\{u_1, u_2, u_3, u_5, u_7\}, \{t_2, t_3, t_4\}, r_2\}$, $\{\{u_1, u_2, u_3, u_5, u_7\}, \{t_2, t_3, t_4\}, r_3\}$, $\{\{u_2, u_3, u_7\}, \{t_1, t_2, t_3, t_4\}, r_2\}$ and $\{\{u_2, u_3, u_7\}, \{t_1, t_2, t_3, t_4\}, r_3\}$. At this level, TRICONS generates a number of candidates by far lower than its competitors, thanks to the generation of tri-generators. The third and final step, *i.e.,* the second call to the $Increase\_Set$ procedure, tends to increase the intent part of each tri-set belonging to $\mathcal{TS}$ in order to extract frequent tri-concepts. For example, the two latter tri-sets merge giving the tri-concept $\{\{u_2, u_3, u_7\}, \{t_1, t_2, t_3, t_4\}, \{r_2, r_3\}\}$ which is added to the set $\mathcal{TC}$. Contrariwise to both CUBEMINER and TRIAS, the tri-concepts are extracted only once. The final result set $\mathcal{TC}$ is then returned by TRICONS which comes to an end with frequent tri-concepts that fulfill the minimum thresholds mentioned above.

5 Experimental Results

In this section, we show through extensive carried out experiment the assessment of the TRICONS[5] performances *vs.* those of TRIAS and DATA-PEELER[6], respectively. We have applied our experiments on two real-world datasets. The first dataset, *i.e.,* DEL.ICIO.US, is considered to be dense, *i.e.,* containing many long frequent tri-concepts at various levels of minimum thresholds values, while the second is considered to be sparse, *i.e.,* containing a large number of tags but only a few of them frequently co-occur in tri-concepts (on average, no more than 2 tags).

• **DEL.ICIO.US: DENSE DATASET**: The DEL.ICIO.US dataset used for our experiments is around 10 MB in size (compressed) and it is freely downloadable[7]. The dense dataset contains 48000 triples : 6822 users, 671 tags and 13102 resources.

• **MOVIELENS: SPARSE DATASET**: The MOVIELENS dataset used is around 13 MB in size (compressed) and it is freely downloadable[8]. The sparse dataset contains 48000 triples : 33419 users, 18066 tags and 13397 resources.

[5] The TRICONS algorithm is implemented in *C++* (compiled with *GCC* 4.1.2) and we used an Intel Core *i*7 CPU system with 6 GB RAM. Tests were carried out on the Linux operating system UBUNTU 10.10.1.

[6] Unfortunately, the code of the CUBEMINER algorithm is not available.

[7] *http://data.dai-labor.de/corpus/delicious/*

[8] *http://www.grouplens.org*

Table 2. Performances TRICONS *vs.* those of TRIAS and DATA-PEELER (in **seconds**) above the DEL.ICIO.US and *MovieLens* datasets

# Triples	Dataset (Type)	TRICONS	TRIAS	DATA PEELER	Dataset (Type)	TRICONS	TRIAS	DATA PEELER
5000		**0,05**	0, 51	638, 22		**0,06**	0, 14	43, 64
15000		**0,55**	0, 91	1538, 15		**0,53**	1, 50	1271, 49
25000	*DEL.ICIO.US*	**3,31**	5, 68	1937, 23	*MovieLens*	**0,91**	3, 30	2010, 77
35000	*(Dense)*	**11,73**	17, 67	2318, 07	*(Sparse)*	**1,51**	6, 34	2909,91
48000		**13,73**	20, 67	2718, 07		**2,69**	11, 52	3851, 38

Performances of TRICONS vs. TRIAS and DATA-PEELER: For mining frequent tri-sets and frequent tri-concepts, we set minimum support values of $minsupp_u = 2$, $minsupp_t = 2$ and $minsupp_r = 1$, *i.e.*, in a frequent tri-concept, at least, 2 users have assigned the same tags (2 at least) to a same resource at least. Table 2 compares the performances (in sec) of the three algorithms above for different values of the number of triples over the mentioned datasets. With respect to the aforementioned minimum support values, the number of the extracted tri-concepts from the DEL.ICIO.US dataset is around 3877. Whereas 1088 tri-conepts are extracted from *MovieLens* dataset.

• **TRICONS** vs. **TRIAS:** For both datasets, the different tests highlight that TRICONS always shows better performances than do TRIAS. For exemple, TRICONS reaches almost $13, 73$ sec when handling 48000 triples from DEL.ICIO.US, showing a drop in execution time of around $33, 57\%$, compared to TRIAS. Moreover, the obtained results, on the both datasets, confirm that this discrepancy between the two algorithms stills in favor of TRICONS as far as the number of triples grows. Interestingly enough, for the sparse dataset, *i.e.,* MOVIELENS, we note, for all values of the number of triples, an average reduction of TRICONS execution time reaching almost $69, 54\%$ compared to TRIAS. The performance differences between these mentioned algorithms can be explained by the fact that TRIAS starts by storing the entire *folksonomy* into main memory before extracting frequent tri-concepts. This memory greedy storage has the drawback to slow the algorithm and alters its execution time as far as the number of triples becomes significant. Contrarily to TRICONS that firstly invokes the FINDMINIMALGENERATORS procedure to extract the tri-generators which constitues the core of the tri-concepts. This specific treatment of TRICONS reduces the memory greediness. Indeed, the number of tri-generators are often by far below the total number of the triples in a *folksonomy*.

• **TRICONS** vs. **DATA-PEELER:** For both datasets and for all values of the number of triples, DATA-PEELER algorithm is far away from TRICONS performances. Indeed, the poor performance flagged out by DATA-PEELER, is explained by the strategy adopted by this later which starts by storing the entire *folksonomy* into a binary tree structure, which should facilitate its run and then the extraction of tri-concepts. Indeed, such structure is absolutely not adequate to support a so highly sized data, which is the case of the *folksonomies* considered in our evaluation. Furthermore, TRICONS is the only one algorithm that does not store the dataset in memory before proceeding the extraction of tri-concepts. In addition, TRICONS generates very few candidates thanks to the clever use of tri-generators that reduce the search space significantly. In contrast, TRIAS and

DATA-PEELER, in addition to store in memory the whole dataset, generate an impressive number of candidates, most of which are stored in memory uselessly given the small number of tri-extracted concepts.

• **TRIAS** vs. **DATA-PEELER:** Contrariwise to experimental results shown in [2], TRIAS outperforms DATA-PEELER since the considered datasets are far away larger. We used real-world datasets similar to those used in [6] which explains why TRIAS is better in terms of performance than its competitor.

6 Conclusion and Future Work

In this paper, we introduced an extension of the notion of closure operator and tri-generator in the *folksonomy* and we thoroughly studied their theoretical properties. Based on these notions, we introduced the TRICONS algorithm, for a scalable mining of tri-concepts, that heavily relies on the order ideal shape of the set of tri-minimal generators. In nearly all experiments we performed, the obtained results showed that TRICONS outperforms the pioneering algorithms of the literature; that is owe to the non-injectivity property of the closure operator. Other avenues for future work mainly address the extraction of other concise representations of frequent tri-sets. In this respect, we will try to expand the steady effort carried within the diadic case towards defining concise representations, *e.g.,* disjunction-free sets (closed) non-derivable sets, (closed) essential itemsets, to cite but a few. It is a thriving issue, since these concise representation have already shown interesting compactness rates [5].

References

1. Besson, J., Robardet, C., Boulicaut, J., Rome, S.: Constraint-based concept mining and its application to microarray data analysis. Intelligent Data Analysis 9, 59–82 (2005)
2. Cerf, L., Besson, J., Robardet, C., Boulicaut, J.: Closed patterns meet n-ary relations. ACM Transactions on Knowledge Discovery from Data 3, 1–36 (2009)
3. Couch, A.L., Chiarini, M.: A Theory of Closure Operators. In: Hausheer, D., Schönwälder, J. (eds.) AIMS 2008. LNCS, vol. 5127, pp. 162–174. Springer, Heidelberg (2008)
4. Ganter, B., Wille, R.: Formal Concept Analysis. Springer (1999)
5. Hamrouni, T., Yahia, S.B., Nguifo, E.M.: Sweeping the disjunctive search space towards mining new exact concise representations of frequent itemsets. Data and Knowledge Engineering 68(10), 1091–1111 (2009)
6. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: Discovering shared conceptualisations in folksonomies. Web Semantics: Science, Services and Agents on the World Wide Web 6, 38–53 (2008)
7. Ji, L., Tan, K.L., Tung, A.K.H.: Mining frequent closed cubes in 3d datasets. In: Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea, pp. 811–822 (2006)
8. Lehmann, F., Wille, R.: A Triadic Approach to Formal Concept Analysis. In: Ellis, G., Rich, W., Levinson, R., Sowa, J.F. (eds.) ICCS 1995. LNCS, vol. 954, pp. 32–43. Springer, Heidelberg (1995)
9. Mika, P.: Ontologies are us: A unified model of social networks and semantics. Web Semantics: Science, Services and Agents on the World Wide Web 5(1), 5–15 (2007)
10. Zaki, M.J.: Closed itemset mining and non-redundant association rule mining. In: Liu, L., Ozsu, M.T. (eds.) Encyclopedia of Database Systems. Springer (2009)

SHARD: A Framework for Sequential, Hierarchical Anomaly Ranking and Detection

Jason Robinson[1], Margaret Lonergan[1], Lisa Singh[1],
Allison Candido[1], and Mehmet Sayal[2]

[1] Georgetown University, Washington, DC 20057, USA
[2] Hewlett Packard, Palo Alto, CA 94304, USA

Abstract. This work explores unsupervised anomaly detection within sequential, hierarchical data. We present a flexible framework for detecting, ranking and analyzing anomalies. The framework 1) allows users to incorporate complex, multidimensional, hierarchical data into the anomaly detection process; 2) uses an ensemble method that can incorporate multiple unsupervised anomaly detection algorithms and configurations; 3) identifies anomalies from combinations of categorical, numeric and temporal data at different conceptual resolutions of hierarchical data; 4) supports a set of anomaly ranking schemes; and 5) uses an interactive tree hierarchy visualization to highlight anomalous regions and relationships. Using both synthetic and real world data, we show that standard anomaly detection algorithms, when plugged into our framework, maintain a high anomaly detection accuracy and identify both micro-level, detailed anomalies and macro-level global anomalies in the data.

Keywords: Anomaly detection framework, multi-resolution anomalies, ensemble method.

1 Introduction

Anomaly detection has many applications, including fraud detection, outbreak identification, and data scrubbing [13] [4]. Each of these domains contains its own semantic relationships, many of which can be modeled as hierarchical. In this paper, we present a framework that allows users to identify anomalies across different levels of these hierarchical structures. For example, in fraud detection, users may be interested in detecting fraudulent behavior across different time granularities (weeks, month, years) or across different locations (neighborhood, city, state). In this case, both time and location are different examples of semantic hierarchies that can be used to identify recurring or aggregated anomalies. Figure 1 shows an example of a sequential, time based hierarchy that we will refer to as an *anomaly tree*. Each level of the anomaly tree represents a different granularity of time. By viewing these different semantic groups of data hierarchically, users can better understand how anomalies propagate through different sequential, hierarchical relationships associated with their applications. Are anomalies scattered or recurring? Are some days, months, or years more anomalous than others?

In this work, we propose $SHARD$, a flexible framework for **S**equential, **H**ierarchical, **A**nomaly, **R**anking, and **D**etection that supports incorporation of hierarchical semantics

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 243–255, 2012.

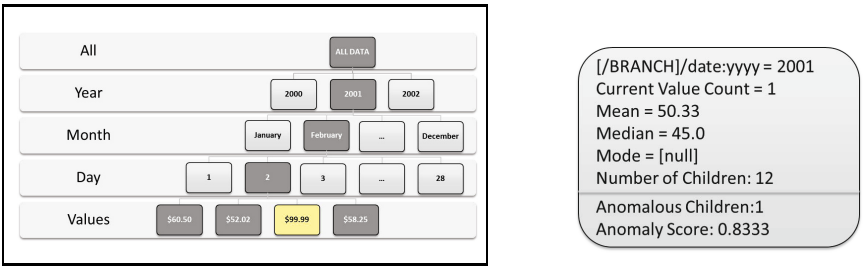


Fig. 1. Anomaly tree example and individual node statistics

across numeric and categorical data into unsupervised, anomaly detection and ranking. This work makes the following contributions. First, we present system and design considerations for developing a general framework for hierarchical anomaly detection. These considerations lead to the decoupling of data formats, outputs, and the definition of 'anomalous' for a given use case. The second contribution is the framework itself, which allows single or multiple anomaly detectors to work together. Most importantly it allows domain experts to drive the anomaly detection process by scripting meaningful, hierarchical relationships between the attributes. Finally, we present experiments on synthetic and real world data sets that show similar performance of detailed, micro-level anomaly detection when compared to the baseline detector performance without the framework; the experiments also demonstrate high-order macro-level anomalies that would completely escape the expert's view without the framework.

The remainder of this paper is organized as follows. Section 2 presents related literature. Section 3 presents background concepts. Our framework is presented in section 4, followed by experimental results in section 5, and the conclusions in section 6.

2 Related Literature

A large body of literature on anomaly detection exists. For a detailed survey of anomaly detection techniques, we refer you to [4] and [13].

Anomaly Detection Frameworks: A few anomaly detection frameworks have been proposed in the literature. For example, Chandola [3] proposes a Reference Based Analysis (RBA) framework for analyzing anomalies with numeric and categorical data in sequences and time series. While RBA offers summary visualizations, it does not offer the multi-resolution evaluations, the interactive visualizations, or the plugin detection and ranking algorithms that our framework does. Nemani *et al.* [12] propose a framework for detecting anomalies in spatial-temporal data. This framework supports plugin detection algorithms; yet, it does not appear to support visualization of multi-granular time series, nor is it clear how customizable other aspects of this framework are.

Anomaly Detection Algorithms: A number of approaches for anomaly detection of time series data exist [5], [8], [10]. Antunes and Oliveira [5] transform the time series into forms that can use standard approaches for anomaly detection. Keogh, Lonardi, and Chiu [10] evaluate the frequency of substrings in a time series and compare the resulting distribution to a baseline time series. Li and Han [11] explore anomaly detection

in multidimensional time series data, identifying the top-k outliers for each detection method and iteratively pruning these sets until a uniform set of anomalies is discovered. All of these sequential anomaly detection algorithms focus on single resolution anomaly detection. Instead, this work focuses on a framework that supports integration of many algorithms across multiple resolutions.

Joslyn and Hogan [9] explore similarity metrics in directed acyclic graphs and other hierarchical structures. Their work can be utilized to visualize and find anomalies in ontologies. While the ideas concerning semantic hierarchies that we present are implicit in Joslyn and Hogan's work, their focus is entirely on similarity metrics in these tree structures and not on the full implementation of an anomaly detection framework.

3 Hierarchical Anomalies

Suppose we are given a data set D, containing a set of attributes or features, $F = \{F_1, F_2, \ldots, F_m\}$, where m is the number of features in D. Each feature contains an ordered list of n values, $F_i = [v_1, v_2, \ldots, v_n]$. We define an anomaly, A, as a data point or set of data points that deviate or behave differently than the majority of *comparison data*, where the comparison data represents values for one or more features in D. We purposely define an anomaly broadly since the type of deviation of interest can vary depending on the data type (numeric, categorical, etc.) and/or the domain characteristics.

Even though our framework can handle any data that can be represented sequentially and hierarchically, including natural language (document, sentences, words, syllables, letters) and genetic sequences (DNA, genes, proteins), for ease of exposition and ubiquity of data, we focus on time series data and time anomaly trees. In this case, data values exists for each feature in the data set at n time points. We also define a set of semantic resolutons $r = \{r_1 ... r_h\}$, where each resolution represents a different semantic grouping for data in D. The semantic groupings for our example in figure 1 are day, month, and year, $r = \{day, month, year, all\}$. These semantic groupings can then be used as the basis for creating a time anomaly tree T of height h, where $h = 4$ for our example. The resolutions tell us the granularity of data associated with each node in a particular level of the tree. The leaf nodes contain statistics about data values at resolution r_1, the day resolution in our example. The parent nodes of the leaf nodes contain statistics about the data values at resolution r_2, e.g. the month resolution, and so on. Given this anomaly tree, we define a hierarchical anomaly $A(n_l)$ to be a node n at level l that deviates significantly from other nodes (or a subset of other nodes) at level l in the anomaly tree, where deviation is measured by one or more detectors selected by the user and significance is algorithm specific.

For example, in a stock data domain, a single company can be considered anomalous if it has an unlikely, sudden surge and subsequent drop in price, if it has an unlikely surge in price that is maintained for some sustained duration, e.g. month, before dropping back to normal, if daily behavior differs drastically from other companies', or if the company manifests a combination of these unusual behaviors. The specific type of behavior identified depends on the detectors and rankers specified by the user.

4 Anomaly Detection Framework

Our high level algorithm for anomaly tree construction and annotation is presented as Algorithm 1. The input to the algorithm is the data (D), an ontology template that specifies the semantic relations of interest (τ), the anomaly detectors of interest (A), and an anomaly ranker (R). Using this information, the framework builds an anomaly tree by assigning data values to the nodes and updating the node summary statistics according to the ontology template, runs different anomaly detectors on the nodes of this tree to obtain a set of anomaly scores for each node, and ranks the anomalies in the tree by computing a score based on criteria such as the level of agreement between the anomaly detectors and the anomaly scores of the child nodes. The resulting tree is then used for an interactive tree visualization that can be analyzed by the user. The remainder of this section describes the framework and different design decisions.

Algorithm 1. Anomaly tree construction and annotation

INPUT: Template τ, Anomaly Detectors A, Ranker R, Data D
OUTPUT: T

function T = BUILD_TREE(τ, D)
function IDENTIFY_ANOMALIES(T, A)
function RANK_ANOMALIES(T, R)
return T

4.1 Ontology Template

The ontological tree template not only decides the hierarchy of where and how feature values are organized and propagated, but also determines how the detectors evaluate nodes. Specific considerations are 1) the range of nodes that maintain summary statistics for the detectors to analyze, 2) normalizing or scaling of multivariate combinations, and 3) sorting of temporal or ordinal features. Table 1 shows an example ontology template and the resulting anomaly tree. The XML template describes an application that attempts to find three different semantic hierarchies based on time, industry, and employee education.

4.2 Anomaly Tree Structure

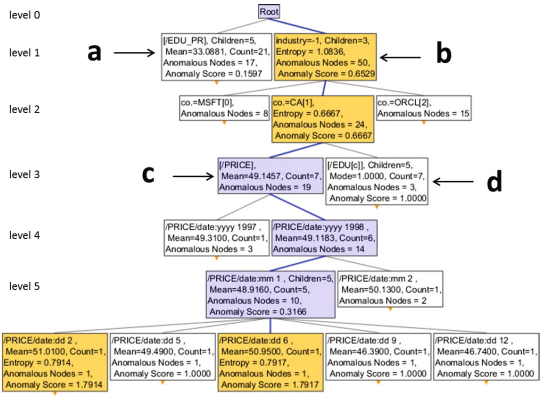
The anomaly tree T generated by the ontology template consists of multiple node types.

Definition 1. *The **leaf nodes** at the lowest level of the tree contain data values. Data from these nodes are aggregated and propagate information to the remaining levels of the tree. **Semantic grouping nodes** are non-leaf nodes that are associated with a feature and group children nodes according to the feature values. **Branching nodes** create a branch of nodes to be evaluated for anomalies. These nodes determine how the child values are evaluated and propagated through T. The propagation of leaf node values stops at the branching node.*

Each node type handles individual data values differently. Semantic grouping nodes split on every new value of the attribute specified in the ontology template. Branching

Table 1. XML template and anomaly tree for XML template. Nodes a, c and d are examples of branching Nodes. Node b is a semantic grouping node, as are all nodes below c and d. Node d also specifies the data propagation to be categorical.

```
<DataTreeTemplate>
   <Node attribute="industry">
      <Node attribute="company">
         <Node attribute="edu" branch="EDU[c]"
               propagateValues="True" >
            <Node attribute="employee">
               <Leaf attribute="edu" />
            </Node>
         </Node>
      <Node attribute="date" step="2"
            branch="PRICE"
            propagateValues="True" >
         <Node attribute="date" step="0">
               <Node attribute="date"
                           step="1" >
                  <Leaf attribute="price"/>
               </Node>
            </Node>
         </Node>
      </Node>
   </Node>
   <Node attribute="edu" branch="EDU_PR"
         propagateValues="True">
      <Node attribute="date" step="2" >
         <Leaf attribute="price"/>
      </Node>
   </Node>
</DataTreeTemplate>
```



nodes are not associated with a value. Instead they store summary statistics of all descendant nodes and tell the detectors whether or not to search for anomalies in a particular branch. The branch creation process creates a root node and a set of children nodes, where each child corresponds to a branching node based on attribute values specified in the ontology template. For example, in the tree path industry/company/[PRICE]/Price/yyyy/mm/dd/price[1], all nodes are grouping nodes except for [PRICE] and the leaf node price data. The leaf nodes propagate their values upward to the top branching node, which means that every parent node is a summary of all of its child nodes. The XML example has two leaf attribute values, price and education that anomalies will be calculated for.

The branch EDU[c] creates a branching node that maintains summary statistics (e.g. $mode$) of the categorical datatype education for each employee in the semantic grouping node $company$, so that we can determine the most frequent level of education per company. Likewise, the parent semantic grouping node $industry$ allows the researcher to also evaluate levels of education across industries. Branching node [EDU_PR] aggregates prices by the average levels of education across all companies.

Table 1 also shows portions of the anomaly tree for the specified XML template. In this example, there is only one industry, technology, under which there are three nodes, one for each of the companies.[2] The arrows at the bottom of nodes indicate nodes that can be expanded to show their children. As the figure illustrates, the anomaly statistics are populated throughout the tree and data statistics from the leaf nodes under a branching node are aggregated as they are pushed up to the branching node, populating the intermediary nodes along the way. Each intermediary node maintains summary statistics of its children nodes. The month level node for the price attribute, for

[1] The XML template in table 1 uses the keyword 'step' to identify which time steps to split on.
[2] See http://cs.georgetown.edu/∼singh/SHARD/ for larger figures, data sets, and source code.

example, maintains the average price for all the children day nodes. Other statistics are also calculated, including median, mode, standard deviation, and entropy.

4.3 Baseline Anomaly Detectors

The anomaly detectors use the anomaly tree, T, to determine the degree of anomalousness of each node in T. This is accomplished by running each user specified anomaly detection algorithm, e.g. statistical significance or entropy, for each element in the tree. Along with the basic detectors, SHARD includes an ensemble detector that combines the detection results of the individual detectors using a weighted voting algorithm, where the weights are prespecified by the user. Once the anomaly scores are computed by the different detectors, the tree nodes are annotated with this additional information. This is also illustrated in Table 1.

In order to identify an anomaly, a data value must be compared to other data values. When evaluating a particular node in T, we use neighboring nodes as comparison data. However, how these nodes are used differs depending on the particular anomaly detection algorithm. For example, table 1 shows the current node under consideration to be day 6 of month 1 (January) of year 1998 of CA, Inc. The options for comparison data for this example include: 1) all immediate sister nodes, all nodes in January for this year and company; 2) all prices for all months under the same company; 3) all prices for all months and companies; 4) all the January 6ths' for the current year across all companies; and 5) the averages of the previous days or months. The SHARD framework includes three parameterized defaults: 1) all local siblings (sister) nodes; 2) all nodes at the same tree height for the same attribute; and 3) previous nodes at the same tree height for the same attribute. Other options can be specified at configuration time and new options are straightforward to integrated into framework.

4.4 Ranking Anomalies

Once all of the detectors have evaluated the nodes in T, the algorithm then runs a user specified ranking method to assign an overall anomaly score to each node. The ranking procedure can compute the anomaly score based on any of the following criteria: 1) the anomaly scores provided by different detectors for a particular node; 2) the percentage of detectors that found a particular node anomalous; 3) the priority of the detectors that found the node to be anomalous; 4) the percentage of child nodes that were found to be anomalous; 5) the importance of the level of granularity in which the anomalous node occurs; and 6) whether anomalies occur in other parallel branches at the same granularity. Our intuition is that the level of anomalousness depends on the domain priorities, objectives and definitions of comparison data. Therefore, we incorporate a tunable ranker that can be adjusted to these considerations. Ranking based on the percentage of anomalous children is the default ranker in SHARD, although we also provide other ranking procedures that combine different subsets of the mentioned factors.

4.5 Anomaly Tree Visualization

SHARD uses the SpaceTree [14] hierarchical visualization application to highlight the most anomalous nodes based on a color heat map. SpaceTree reads in XML and

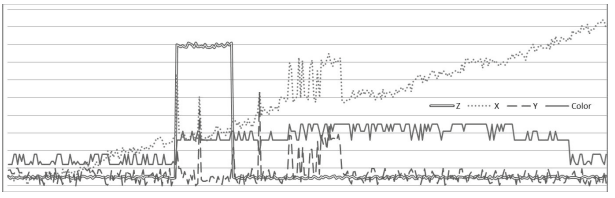


Fig. 2. One year of synthetic time series data

displays an interactive tree of variable depth and width. This interactive software enables users to expand the entire tree or focus on subtrees of different branches of the full tree while hiding other subtrees. Doing this helps the user see where anomalies occur across multiple resolutions. Because our framework is customizable, any amount of detail can be displayed for each node including ranking scores, statistical summaries, individual detector results, and raw data. This interactive visualization supports both an overview and a detailed view, allowing for a more comprehensive analysis of the anomalies. Most of the tree images in this paper were generated using SpaceTree.

5 Empirical Evaluation

In this section, we evaluate our framework on synthetic and real world data sets. Our evaluation of the SHARD framework focuses on detection accuracy and anomalies discovered. Specifically, we compare the accuracy of the detectors outside our framework with the same detectors within the SHARD framework and show that the overall accuracy is generally maintained, while also offering bigger picture insights. We also discuss these insights at different levels of the anomaly tree and demonstrate the flexibility of our framework.

We experimented with four standard anomaly detection algorithms in our framework: 1) the Shewhart algorithm [1], which flags anomalies that are x standard deviations away from the mean; 2) the Cumulative Sum (cusum) algorithm, which tracks the mean of all previous elements and compares the values to the current element; 3) entropy (applied to anomalies as described in [7]); and 4) a thresholding version of Bruenig *et. al*'s [2] Local Outlier Factor (LOF).

The ranking algorithm used in all of the experiments is *RankerA*. This ranker first evaluates the children nodes. If at least half are anomalous, the current (parent) node is also considered anomalous. Otherwise, the sum of all anomaly scores, one from each detector, of a node is divided by the number of children nodes.

5.1 Synthetic Data Experiments

For this analysis, we generated three time series with a numeric data value for each day over a six year period, and one categorical times series. Figure 2 shows each of the numeric time series for a one year period. As illustrated in the figure, each time series has different properties and anomalies. Time series X increases in overall magnitude over time with burst anomalies for 200 random days, one random month of the year (this

Detector	Attribute - Path	Precision	Recall
Shewhart	x - yyyy/mm/dd	75.3%	11.6%
	x - leaf	100.0%	8.9%
	y - yyyy/mm/dd	93.3%	12.5%
	y - leaf	100.0%	54.5%
	z - yyyy/mm	83.3%	45.5%
	z - leaf	3.3%	50.0%
	x,y,z - yyyy/dd	100.0%	2.2%
OVERALL		52.9%	12.7%
Entropy	x - yyyy	50.0%	100.0%
	x - yyyy/mm	12.5%	50.0%
	x - yyyy/mm/dd	29.6%	86.2%
	x - leaf	21.6%	63.0%
	y - yyyy	100.0%	100.0%
	y - yyyy/mm	60.0%	100.0%
	y - yyyy/mm/dd	29.1%	85.7%
	y - leaf	100.0%	100.0%
	z - yyyy/mm	83.3%	45.5%
	z - yyyy/mm/dd	0.4%	25.0%
	z - leaf	3.3%	50.0%
	color - yyyy	25.0%	100.0%
	color - yyyy/mm	6.7%	25%
	color - leaf	30.2%	95.0%
	x,y,z - yyyy	50.0%	100.0%
	x,y,z - yyyy/mm	100.0%	52.9%
	x,y,z - yyyy/mm/dd	33.9%	88.0%
OVERALL		28.3%	82.2%
LOF(1)	x - yyyy/mm/dd	92.0%	41.1%
	y - yyyy/mm/dd	100.0%	29.9%
LOF(3)	x,y,z - yyyy/mm/dd	98.1%	46.4%
OVERALL		95.6%	7.9%

Fig. 3. Single detectors

Detector	Parameters	Attribute	Precision	Recall
Shewhart	thresh=2	x	100.0%	8.9%
	"	y	100.0%	54.5%
	"	z	3.8%	50.0%
	"	color	n/a	n/a
	"	x,y,z	100.0%	10.3%
OVERALL			52.6%	24.2%
Entropy	thresh=1.2	x	21.6%	63.0%
	"	y	100.0%	100.0%
	"	z	3.8%	50.0%
	"	color	1.3%	47.8%
	"	x,y,z	21.8%	62.8%
OVERALL			20.5%	74.3%
LOF	k=15, dim=1	x	9.0%	0.5%
	"	y	100.0%	5.4%
	"	z	0.0%	0.0%
	"	color	n/a	n/a
	k=15,dim=3	x,y,z	85.0%	15.2%
OVERALL			74.6%	6.7%

(a) Baseline detectors

Detector	Attribute - Path	Precision	Recall
Shewhart Entropy LOF(1) LOF(3)	x - yyyy/mm/d	88.2%	43.3%
	x - leaf	100.0%	8.9%
	y - yyyy/mm/d	97.1%	30.4%
	y - leaf	100.0%	100.0%
	z - yyyy/mm	83.3%	45.4%
	z - leaf	3.3%	50.0%
	x,y,z - yyyy/mm/d	99.1%	46.4%
OVERALL		68.4%	25.6%

(b) Ensemble detectors

Fig. 4. Baseline and ensemble detectors

includes several of the random anomalous days), and one random year (this includes approximately 1/3 of its days being anomalous). Time series Y is similar except that the "normal" comparison values across all 6 years remain relatively steady. Like X, it contains randomly anomalous days, months and a year- most of which coincide with the anomalies in time series X. Time series Z is mostly independent of the other two time series and illustrates a plateau anomaly that starts and ends with anomalies found in X and Y. It contains the same anomalous month each year in which all values during this month are consistent for this month, but still much higher than the normal day value for the rest of the year. At the individual day level, the only anomalies are the first day of this month when the values increase and the first of the following month when the values decrease back to normal. We also include a categorical attribute, $Color$, that is dependent on the season in the times series (during months *11,12, 1, 2, 3* {*blue, green, purple*}; *4, 5, 10* {yellow, orange}; and *6, 7, 8, 9*{red, orange, yellow}). An anomalous instance is an out-of-season color that corresponds with the Y anomalies' time points.

Our ontology template for this data set consists of 5 branches underneath the root. The first three simply aggregate each of the continuous variables by year, month and day independently:

[DATE-X]/yyyy/mm/dd/x, [DATE-Y]/yyyy/mm/dd/y, [DATE-Z]/yyyy/mm/dd/z

The fourth branch groups all three variables under each unique date:

[DATE-XYZ]/yyyy/mm/dd/x,y,z

Here, the time series are evaluated together, in the context of each other. In other words, the most anomalous time periods are when all three time series have anomalous behavior during the same time period. Note that there are parameters in the XML to normalize or scale multiple values under a single node. In this run, the configuration was set to Normalize. The final branch, [COLOR][c]/yyyy/mm/color organizes the categorical colors by month and year to capture anomalies in the context of different seasons.

Table 2. Anomaly detectors on the CalIt2 dataset, (dd = day of month; Day = day of week)

Detector	Attribute - Path	Precision	Recall
Shewhart	mm/dd/hh	21.7%	49.4%
	mm/dd/hh/c	25.0%	58.6%
	hh	100.0%	9.1%
	hh/Day/c	24.9%	43.2%
	hh/Day/c/id/c	25.0%	56.5%
OVERALL		24.7%	51.9%

Detectors	Attribute - Path	Precision	Recall
Shewhart Entropy LOF(1)	mm/dd/hh/c	72.2%	4.5%
	Day	50.0%	4.2%
	Day/c	62.8%	4.9%
	Day/c/id/c	60.7%	5.5%
OVERALL		63.9%	4.6%

Detector	Attribute - Path	Precision	Recall
Entropy	mm/dd/hh/c	72.2%	4.5%
	hh/Day	14.3%	58.3%
	hh/Day/c/id/c	60.1%	5.8%
OVERALL		39.5%	4.1%

Detector	Attribute - Path	Precision	Recall
LOF	hh/Day	50.0%	16.7%
	hh/Day/c	61.8%	4.9%
OVERALL		59.5%	1.3%

These various branches show the flexibility of the framework for handling different feature combinations that the user wants to investigate.

Figure 4(a) shows the scores of the baseline algorithms outside of our framework. The algorithms process each attribute individually and flag individual values as being anomalous, but give no indication of anomalous months or years. Figure 3 shows the results of the baseline algorithms within our framework. The overall scores are comparable with the record level scores outside of our framework in figure 4(a); however, a richer picture is gained using our framework: Shewhart now correctly identifies z's anomalous months with much higher accuracy, entropy performs well at nearly all resolutions of the anomaly tree, and LOF's recall is higher for most variables. Finally, figure 4(b) shows the results of the ensemble of these detectors. While the overall accuracy and precision is lower than the single detectors in the framework, the interior nodes of the tree have similar or better precision and accuracy results, demonstrating a potential benefit of a diverse set of detectors for hierarchical anomaly detection.

5.2 Event Attendance Data Results

We now consider an event data set, the CalIt2 dataset [6], for detecting anomalous events. This data set contains two observation time series (people flowing in and people flowing out of the building) over 15 weeks from July to November. There are 48 time points per day. The 'normal' behavior of this data set is a periodic, light flow of people going in and out of this building. When a conference is occurring, the flow increases for what is considered normal at that day and time, and an anomaly occurs.

Using the SHARD framework we specified two parallel branches in the ontology template, which offers two different views of the data. The first is Month/Day/Hour/Count - the intutive hierarchy. The second branch is Hour/DayOfWeek/Count/id/Count. This branch first establishes normal data behavior of the 24 hours of the day across the entire dataset, and then sub-aggregates the data by the day of the week and then the counts. So, it might establish that the average count for 9:00 am is 3.5 people, and the average for 9:00 am/Wednesday is 5.0 people. The next groupings id/Count, then establish counts based on individual records.

Inside the SHARD framework with this XML configuration, Shewhart with a threshold of 1 scores 24.7% precision, 51.9% recall on the anomaly tree nodes; Entropy with

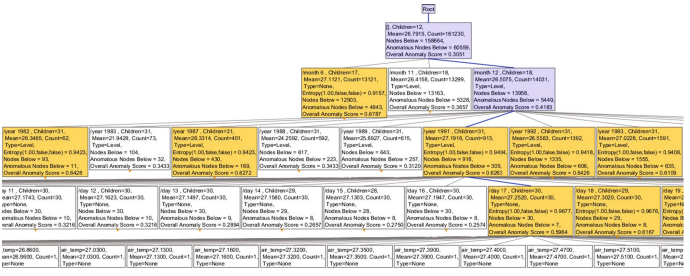


Fig. 5. El Nino anomaly tree: inverted month-year hierarchies. Anomalous nodes shaded orange.

a threshold of 7.5 scores 39.5% precision, 4.1% recall; LOF where k=5 scores 59% precision and 1.3% recall; and these three detectors in an ensemble configuration score 63.9% precision and 4.6% recall. Outside of the SHARD framework Shewhard and Entropy perform comparably on the flat data (pr= 24.8%, re=56.3%, and pr=55.7%, re=5.4%, respectively), but LOF scores 0% precision and recall.

We offer a few observations. First, the 0 score of LOF outside of our system is probably due to at least k records with high counts that are not known events. As these points are considered normal comparison data, no points are flagged anomalous when the comparison data consists of all records. In our framework this happens less because these normal high-count records are dispersed throughout different parts of the anomaly tree. Second, the ensemble run of these three methods produced a higher precision level than any of these three algorithms independently. Third, the SHARD framework produced insight into many different levels of the anomaly tree. Specifically, investigating the SpaceTree nodes that were flagged anomalous, we determined: November is anomalous because it has no events but very high counts, August is anomalous because it has more events than the other months, all Saturdays are anomalous because they do not have any events, one Sunday is anomalous because it is the only Sunday with an event, and three days are anomalous because they are the only days with multiple events.

5.3 Climatology Data Results

Here we use a data set collected by the Pacific Marine Environmental Laboratory to study the El Nino and La Nina phenomena [6]. This data set contains climatology data from 1980-1998, during which there were 6 El Ninos (**1982**, 1987, **1991**, 1992, 1994, **1997**) and 1 La Nina (**1988**). The years in bold were considered very strong. The most anomalous months with unusually high temperatures are typically December of that year and January of the following year. There were 178,080 total readings of date, location, trade winds, humidity and air and sea surface readings.

Using the SHARD framework, we create an XML template that contains a typical, sequential date hierarchy year/month/day/{attribute} structure for each attribute. Using Entropy, threshold=1, the framework flags the appropriate El Nino and La Nina years with 87.5% precision and 58.3% recall using the ocean surface temperature; 88.9% and 66.7%, respectively, with the air temperature readings. Because we do not have ground truth weather information to accurately label all anomalous months and days, the precision and recall cannot be reported for the other levels of the anomaly tree.

Table 3. Anomalous years in stock data set

Industry	Years
Application Software	1999, 2000
Asset Management	2006 - 2008
Beverages - Brewers	2006 - 2008
Investment - Brokerage - Nat.	2000, 2006, 2007
Major Airlines	1998 - 2001, 2006, 2007
Regional Banks	2004 - 2007
Regional Airlines	1999, 2001, 2006

We pause to mention that this data set contains many missing values since not every buoy was equipped to measure all of these attributes. Our framework can handle missing values by creating tree nodes only for values that are present and then searching for local anomalies within the tree.

Because of the flexibility of our XML templating, we also considered an alternative XML template that inverts months with years, so that the hierarchy is month/year/day as shown in figure 5. This means that for the month of December we have all year nodes as children and under each year node all December day measurements. This gives the researcher a very easy way to learn during which years December was most anomalous. Using this inverse technique, if we examine December, we find 85.7% precision and 77.7% recall at tagging the appropriate years. More interestingly, though, the highest ranked nodes correspond very well to the 'strong' El Nino years.

5.4 Stock Data Results

In these experiments, we analyze the NASDAQ daily stock quotes from 1998-2009 of 34 companies in the Technology, Financial, Services and Consumer Goods sectors. There is 1 date attribute, 7 numeric attributes and 5 categorical attributes for 14,805 records. We chose these years and industries because much happened in this decade: there was the dot.com bubble, followed by a correction year, 9/11, and another correction year following the real estate bubble. With the stock data we decided to study the most anomalous years by industry with the XML template configured as Industry/Year/Company Size/Company Name/Month/Day/Closing Price. We again used a default Entropy detector with a threshold of 1. A brief summary of these results can be found in table 3. Although we found the correlations between anomalies in Asset Management and those in Beverages - Brewers unexpected, the rest of the results seem easily interpretable, Application Software's dot.com boom and correction are rightly noted, the airlines show up in 2001, and many financial anomalies start to show up in 2004-2008. These results are consistent with expectations.

5.5 Discussion

The experimental results demonstrate the utility of having a hierarchical anomaly detection framework. Our synthetic and event attendance detection results indicate that the ensemble method has fewer false positives than the individual detection methods and

a higher accuracy than any of the individual methods. We believe this results because the ensemble method is able to capture a more robust image of the data, whereas the individual algorithms are more suited to detect a particular type of anomaly.

Our results also show that the existence of anomalies at one granularity is not indicative of anomalies in other granularities. Figure 5 depicts a feature with many anomalous leaf nodes, but the parents of these nodes are not anomalous as indicated by 'Anomalous Nodes Below'. This is consistent with our understanding of point and contextual anomalies, and that one does not imply the other. Higher granularities are more descriptive of contextual anomalies, and not simply single point anomalies.

Using the SpaceTree application, we were also able to visualize our results in a meaningful way. The user is able to access relevant statistics about each node, as well as quickly see where anomalies are occurring. This is important in our work as mentally visualizing anomalies at multiple granularities is not an intuitive task.

6 Conclusions and Future Work

This work introduces SHARD, a framework that supports analysis of complex, multi-dimensional, hierarchical anomalies. Our framework is robust and allows for easy customization for different applications, as well as easy extensions for adding additional anomaly detectors and rankers. Using our prototype system, we illustrate both the flexibility and utility of this framework on both synthetic and real world data sets. Future work includes expanding the detectors in the framework, allowing for streaming analysis, demonstrating other semantic hierarchies that are not time based, and reducing the number of user specified parameters. Finally, many of the hierarchical aggregates mentioned are examples of cuboids. Extending our tree framework to a cube framework is another promising direction.

References

1. Barnard, G.A.: Control charts and stochastic processes. Journal of the Royal Statistical Society B21, 239–271 (1959)
2. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. SIGMOD Record 29, 93–104 (2000)
3. Chandola, V.: Anomaly detection for symbolic sequences and time series data. PhD thesis, University of Minnesota (2009)
4. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computer Surveys 41(3), 1–58 (2009)
5. Oliveira, A.L., Antunes, C.M.: Temporal data mining: An overview. In: KDD Workshop on Temporal Data Mining (2001)
6. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
7. He, Z., Deng, S., Xu, X.: An Optimization Model for Outlier Detection in Categorical Data. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005, Part I. LNCS, vol. 3644, pp. 400–409. Springer, Heidelberg (2005)
8. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artificial Intelligence Review 22, 85–126 (2004)
9. Joslyn, C., Hogan, E.: Order Metrics for Semantic Knowledge Systems. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) HAIS 2010, Part II. LNCS, vol. 6077, pp. 399–409. Springer, Heidelberg (2010)

10. Keogh, E., Lonardi, S., Chiu, B.: Finding surprising patterns in a time series database in linear time and space. In: ACM KDD, pp. 550–556. ACM (2002)
11. Li, X., Han, J.: Mining approximate top-k subspace anomalies in multi-dimensional time-series data. In: VLDB, pp. 447–458. VLDB Endowment (2007)
12. Nemani, R., Hashimoto, H., Votava, P., Melton, F., et al.: Monitoring and forecasting ecosystem dynamics using the terrestrial observation and prediction system (tops). Remote Sensing of Environment 113(7), 1497–1509 (2009)
13. Patcha, A., Park, J.M.: An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks 51(12), 3448–3470 (2007)
14. Plaisant, C., Grosjean, J., Bederson, B.B.: Spacetree: Supporting exploration in large node link tree, design evolution and empirical evaluation. In: IEEE Symposium on Information Visualization, p. 57 (2002)

Instant Social Graph Search*

Sen Wu, Jie Tang, and Bo Gao

Department of Computer Science and Technology,
Tsinghua University, Beijing, 100084, China
{ronaldosen,elivoa}@gmail.com, jietang@tsinghua.edu.cn

Abstract. In this paper, we study a new problem of instant social graph search, which aims to find a sub graph that closely connects two and more persons in a social network. This is a natural requirement in our real daily life, such as "Who can be my referrals for applying for a job position?". In this paper, we formally define the problem and present a series of approximate algorithms to solve this problem: Path, Influence, and Diversity. To evaluate the social graph search results, we have developed two prototype systems, which are online available and have attracted thousands of users. In terms of both user's viewing time and the number of user clicks, we demonstrate that the three algorithms can significantly outperform (+34.56%-+131.37%) the baseline algorithm.

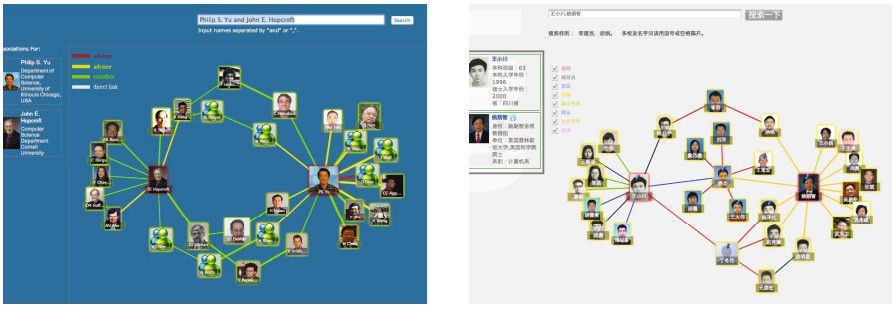
1 Introduction

With the big success of many large-scale online social networks(e.g., Facebook, RenRen, MySpace, Ning, and Twitter) and the rapid growth of the mobile social networks (e.g., FourSquare, Data.net, Strands), there has been a large increase in the people's social friends especially online social network friends. The online social network is becoming one of the most important ties between people's daily life and virtual web space. For example, Facebook, which is the most-visited site on the web, contains more than 600,000,000 unique visitors(users) since Jan 2011; Foursquare, a location-based mobile social network, has attracted 6 million registered users by the end of 2010. There is little doubt that most of our friends are online now.

In such a case, one important requirement in the social network is to find the connections (also called associations) among persons [14], which has many direct applications. For example, to find referral people for applying for a job position [9]. Indeed, LinkedIn has a very important function, which allows users to see how far (how many degrees) you are from another user and allow users to write recommendation to a friend. In particular, interesting questions arise: "Who are the good referrals for me to apply for the PhD program of a university?", "What are my relationships to the Turing Award winner, Prof. John Hopcroft?", and "Who are the experts on topic X and how to connect him/her?". For all the questions, the answers should be returned in real time. The general problem is referred to as instant social graph search. Please note that the connection between people might be directed, e.g., via a coauthorship; or indirected, e.g., the friend's friend.

* The work is supported by the Natural Science Foundation of China (No. 61073073) and Chinese National Key Foundation Research (No. 60933013, No. 61035004), a special fund for Fast Sharing of Science Paper in Net Era by CSTD.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 256–267, 2012.



(a) Coauthor network (b) Alumni network of a university

Fig. 1. Two examples of instant social graph search in a coauthor network and a university alumni network. The left figure shows the social graph between two computer science experts: "Philip Yu" and "John Hopcroft" in the coauthor network. The right figure shows the social graph between "Andrew Yao" (Turing Award winner) and "Xiaochuan Wang" (Vice President of a company) in the alumni network.

Motivating Example. To clearly motivate this problem, Figure 1 gives examples of instant social graph search on a coauthor network and an alumni network of a university. The figure 1(a) shows the social graph between two experts in computer science: "Philip Yu" and "John Hopcroft" and the figure 1(b) plots the social graph between one faculty "Andrew Yao" (Turing Award winner) and one alumnus "Xiaochuan Wang" (Vice President of a company) discovered from the alumni network. In the figure 1(b) different colored links indicate different types of relationships. For example, in the left figure, yellow-colored link indicates advisee relationship, red-colored link indicates advisor relationship, and green-colored link indicates coauthor relationship. While in the right figure, the types of relationships include: advisor, colleague, classmate, high-school alumni, friendship, etc. "Pictures Worth a Thousand Words". We can see such a social graph is very helpful to understand the social connection among persons. With such a graph, we can easily find trusted referrals for connecting a person (e.g., an expert), who are very likely to give a help because they are friends of your friends.

The problem is non-trivial. One fundamental challenge is how to effectively select and generate the social graph between (or among) persons in real time. It is well-known that any two persons in the world are connected in six steps or fewer [13]. This means that almost any persons in the world are within your six-degree social circle. At the same time, this also implies that for any two persons, the number of connections between them would be huge. Obviously it is infeasible to display all the connections between persons in a social graph. Our preliminary study shows that when a graph consists of more than 50 nodes, the user will have difficulties in understanding the meaning of the graph, and quickly lose interest to the graph (with less viewing time).

Challenges and Contributions. In this work, we try to conduct a systematic investigation of the problem of *instant social network search*. The problem poses a set of unique challenges:

- *Goodness.* How to quantify the goodness of a sub network among people? Specifically, given a graph G and a query consisting of multiple person nodes in the graph, how to find a "good" subgraph of G that contains the query nodes.
- *Diversity.* How to diversify the returned graph so that it captures the whole spectrum of the connections among the queried persons? It is widely realized that diversity is a key factor to address the uncertainty in an information need [1,21].
- *Efficiency.* How to return the queried graphs instantly? As real social networks are getting larger with millions or billions of nodes, it is necessary to design an efficient algorithm which can return the queried social graphs in (milli-)seconds.

To address the above challenges, we first precisely define the problem and then propose an efficient algorithm to solve the problem. We further incorporate the topic diversity into the objective function and propose an enhanced diversity algorithm. We have developed two prototype systems, one is for a coauthor network and the other is for a university alumni network, both of which are online available and has attracted thousands of users. We evaluate the performance of the proposed algorithms in terms of user viewing time and number of user clicks. Experimental results on one-month query log show that the proposed algorithms can significantly outperform (+34.56%- +131.37% in terms of viewing time) the alternative baseline algorithm. We also find that the Diversity algorithm achieves the best performance. Our experiments also validate the efficiency of the presented algorithms, which can return the search results for most queries in 2 seconds.

2 Problem Definition

In this section, we first give several necessary definitions and then present the problem formulation.

A social network is modeled as an undirected graph $G = (V, E, U, W)$, where V represents a set of users, $E \subset V \times V$ represents a set of social relationships between users, $u_i \in U$ represents the importance (or activity) of user v_i, and $w_{ij} \in W$ represents the closenesses between user v_i and user v_j. Given a query of k persons $q = \{v_{q1}, \cdots, v_{qk}\}$, the goal is to find a set of users $S_q \subset V$ to closely connect the queried users in q, by considering the *importance* of nodes, the *closeness* of relationships, and the *connectedness* to the query users. In different networks, the three criteria can be instantiated in different ways. For example, in a coauthor network, importance can be defined as the number of papers published by the author (or the total number of citations of the author, or simply the value of H-index [7]), while the relationship's closeness can be defined as the number of coauthored papers. Formally, we can define the social graph search problem as follows:

Definition 1. ***Social Graph Search****: Given a social network $G = (V, E, U, W)$ and a query $q = \{v_{q1}, \cdots, v_{qk}\}$ of k persons, the goal of social graph search is to find a subgraph G_q of G, such that (1) G_q contains the queried persons, i.e., $\{v_{q1}, \cdots, v_{qk}\} \subseteq V_q$, (2) nodes in the subgraph G_q are closely connected, and (3) the number of nodes in the returned graph is less than a threshold, i.e., $|V_q| \leq M$.*

In the definition, we explicitly constrain the number of persons in the returned social graph as M (condition (3)). This constraint is necessary for controlling the size of the returned subgraph; otherwise, algorithm would trivially return the whole social graph. Now the problem is how to satisfy the second constraint: nodes in the subgraph G_q are closely connected, more specifically, how to quantify the connectness of a graph. To make things simple, we define the connectness as the number of relationships among the selected nodes in the graph G_q. Another challenge is how to diversify the selected nodes in the graph. In Section 3 we will introduce how we achieve these two goals and find the trade-off balance between them.

Several relevant research efforts [2] has been made so far. However, our problem addressed in this paper is very different from existing work. For example, [2] proposes the notion of semantic association and has investigated how to rank the semantic associations based on the information gain. However, association search is different from social graph search. The former is to find association paths to connect two persons, while our goal is to find a social graph to connect multiple persons. Our problem can be viewed as a generalized problem of the association search. Faloutsos et al. [5] also study how to efficiently discover a connection subgraph between nodes in a graph. However, they do not consider the importance of nodes and weight of relationships together, and they do not give an objective method to evaluate the discovered subgraph. Our work aims at satisfying both of the two goals: relevance and diversity. Sozio and Gionis [15] study a community-search problem, which has an objective similar to our work. However, the algorithm cannot be scaled up to handle networks of millions of nodes in real time.

3 Algorithms

The problem of social graph search as we defined in Section 2 is NP-hard, which can be proved by a reduction to the Dominating Set Problem. In this section, we will introduce three algorithms to obtain approximate solutions of the problem, respectively called Path, Influence, and Diversity. For easy explanation, we consider only two persons in the query, i.e., $q = \{v_{q1}, v_{q2}\}$.

3.1 Basic Ideas

There are two basic objectives we want to achieve in the social graph search problem. The first is to find important nodes and the second is to find nodes that could closely connect the queried nodes. In general, the connective social graph between user v_{q1} and v_{q2} can be decomposed into multiple paths between them [8]. Therefore our first idea is to cast the problem as shortest associations finding. According to the weighted importance w_{ij} between users, we can find the shortest association path between any two users using dynamic programming, and then find the top-k shortest paths by relaxing the search condition. This algorithm is called *Path*. It is efficient and easy to implement. However, the algorithm does not consider the importance of nodes and also the possible redundant information (i.e., the same nodes and edges) between different paths.

Input: G, number of selected pathes k, bound to shortest path δ;
Output: S;

Initialize $S = \varnothing$;
Initialize $D = \inf$;
Use Dijkastra algorithm to calculate the shortest path D;
for $i = 1$ *to* $D + \delta$ **do**
    create a queue Q;
    enqueue source on Q;
    mark source;
    while Q *is not empty* **do**
        dequeue an item from Q into V;
        foreach *edge* e *incident on* v *in Graph* **do**
            let w be the other end of e;
            if w *is not marked:* **then**
                mark w;
                enqueue w onto Q;
            end
        end
    end
end
Set all the marked node on the path in S;
Output S;

Algorithm 1. Path algorithm

We therefore propose an influence maximization based algorithm, called *Influence*. The idea is to cast the problem as that of influence maximization [10], whose goal is to find a small set of nodes in a social network that maximize the spread of influence under certain models. To further consider the diversity, we propose an enhanced algorithm called *Diversity*. The basic assumption is that each user may focus on different aspects (topics). Without considering the diversity, the resultant graph may be dominated by a major topic (e.g., a resultant graph from the alumni network may be dominated by one's classmates). The new algorithm incorporates the topic information into an objective function, thus the selection strategy achieves a trade-off between the influence of the selected nodes and the diversity of all topics over the resultant graph.

3.2 The *Path* Algorithm

A straightforward method to deal with the instant social graph search problem is to find the shortest paths between two persons and then use those persons appearing in the paths to construct the social graph. We called this baseline algorithm as Path. More specifically, we take the negative weight $-w_{ij}$ of each edge $e_{ij} \in E$ in the network G as its distance. By using a (heap-based) Dijkastra algorithm [4], we can obtain the shortest path from all nodes to a target node in the network, with a complexity of $O(n\log(n))$. Then we use a depth-first (or width-first) search to find near-shortest pathes by bounding the length (distance) of the path within a factor (i.e., $\leq (1 + \delta)$) of the shortest path. The algorithm is summarized in Algorithm 1.

Limitations. The *Path* algorithm does not consider the correlation (dependency) between two paths, thus it is very likely to choose two "redundant" paths (i.e., paths sharing a number of common nodes). Actually, in our data sets, analysis shows that in many cases, the top 10 shortest paths only have one or two node(s) difference. Another limitation of the algorithm is that it does not consider the importance of each node.

3.3 The *Influence* Algorithm

Our second idea is to cast the social graph search problem as that of influence maximization [10], whose goal is to find a small set of nodes in a social network that maximize the spread of influence under certain models.

In order to achieve this, we first translate the social network into an influence graph where each node indicates a path between the queried nodes. If two paths have a common node, we create an edge between the corresponding nodes in the influence graph and the weight of the edge is the number of common nodes of the two paths. It is easy to know that the new influence graph is a connected graph and then we employ a greedy algorithm [3] to select the nodes in the new graph (i.e., paths in the original graph). The algorithm is based on the Monte Carlo random process. It runs iteratively and in each round, the algorithm selects one vertex into the selected set S such that this vertex together with the current set S maximizes an influence score. Equivalently, this means that the vertex selected in round i is the one that maximizes the incremental score of influence. To do so, for each vertex v that is not in S, the influence spread of $S \cup v$ is estimated with R repeated simulations of random process. The algorithm is presented in Algorithm 2.

Limitations. The *Influence* algorithm considers the network information, and it can avoid redundant nodes (nodes are close with each other in the transferred graph), by adopting a degree discount method [3]. However, it does not consider the diversity problem. In some extreme cases, one major aspect (topic) may dominate the resultant graph. This leads us to propose the *Diversity* algorithm.

3.4 The *Diversity* Algorithm

On a social network, each user may have interest (or expertise) on multiple different topics. When the user searches for social graphs between two persons, he is not only interested in the network that closely connects the two persons, but also interested in how the two persons are connected on different aspects. For example, when the user searches for the social graph between two professors respectively from data mining and theory. The user might be interested in knowing how the two professors build collaborations in different fields.

Hence, we augment the social network model with topic representation, i.e., $G = (V, E, U, W, R)$, where $\mathbf{r}_i \in R$ is a vector denoting the topic distribution of each user v_i with each element r_{ij} representing the probability of user v_i's interest (or expertise) on topic j. Please note that the diversity problem can be also defined in some other ways. For example, we can consider different social ties and thus expect the returned social graph contain diverse social ties. According to the definition, the social graph search problem with diversity can be re-defined as to find a small subset of users to *statistically* represent the topic distribution of the social graph between the queried persons.

Input: G, number of selected pathes k;
Output: S;

Initialize $S = \varnothing$;
Initialize $R = 20000$;
for $i = 1$ *to* k **do**
  foreach *vertex* $v \in V \backslash S$ **do**
    $s_v = 0$;
    for $j = 1$ *to* R **do**
      $s_v += |RanCas(S \cup \{v\})|$;
    end
    $s_v = s_v / R$;
  end
  $S = S \cup \{argmax_{v \in V \backslash S}\{s_v\}\}$;
end
Output S;

Algorithm 2. Influence algorithm

The proposed *Diversity* algorithm is based on two principles that are used to select representative users in our physical social network: *synecdoche* (in which a specific instance stands for the general case) and *metonymy* (in which a specific concept stands for another related or broader concept) [12]. Thus one problem is how to define the topic-based representative degree between users. Without loss of generality, we define the representative degree of user v_i on v_j for topic z according to the similarity between two persons on the topic, i.e.,

$$rep(v_i, v_j, z) = \frac{|r_{iz} - r_{jz}|}{r_{iz}} \tag{1}$$

Therefore, our objective is to select a set S of persons who can best represent all the other persons in the social graph on all topics, formally we can define the following objective function:

$$\mathcal{O}(S) = \max_{v_i \in S} \sum_{z} \sum_{v_j \in V \backslash S} rep(v_i, v_j, z) \tag{2}$$

Maximizing the representative degree on all topics is obviously NP-hard. Some trade-offs should be considered as we may need to choose some less representative nodes on some topics to increase the total representative degree on all topics. We give a greedy heuristic algorithm. Each time we traverse all candidate persons in the social graph and find the individual that most increases the representative function $\mathcal{O}(S)$. To increase in representative function achieved by adding a person $v_i \in V$, we only need to consider the topics that v_i can mainly contribute to ($r_{ik} > 0$) and all v_i's neighbors (we say v_j is v_i's neighbor if $rep(v_i, v_j, z) > 0$ for some $v_j \in V \backslash S$). The algorithm is summarized in Algorithm 3:

4 Experimental Results

For evaluation, we have deploy the presented algorithms in two systems: a social graph search in Arnetminer[1] [19] and an alumni network system.

[1] http://arnetminer.org

```
Input: G, number of selected pathes k;
Output: selected users S;
S = ∅;
while |S| < k do
    max = −1;
    foreach v_i ∉ S do
        foreach r_iz > 0 do
            foreach v_j ∈ G that rep(v_i, v_j, z) > 0 do
                Compute the increment of O(S ∪ v_i) − O(S) on topic z;
            end
            Compute the total increment;
        end
        if increment > max then
            v = v_i; Update max;
        end
    end
    S = S ∪ {v};
    Update O(S).
end
Return S;
```

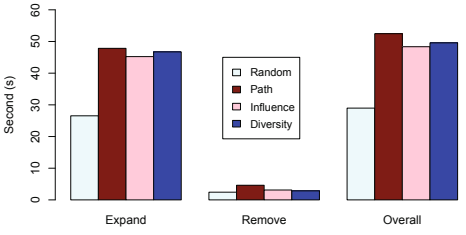
Algorithm 3. Diversity algorithm

4.1 Experiment Setup

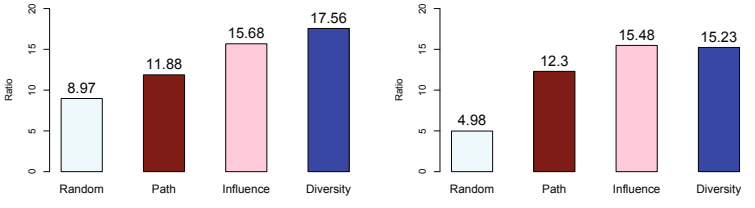
Data Sets. We perform our experiments on the two systems which contain two different data sets: coauthor network and alumni social network.

- Coauthor network. In the coauthor network, we focus on studying the coauthor social graph, which consists of 1,483,246 authors and 47,443,857 coauthor relationships. We also employ a time-dependent factor graph model [22,20] to discover the advisor-advisee relationships from the coauthor network. The social graph search function has been integrated into academic analysis and mining system for a few months, and attracted tens of thousands of accesses.
- Alumni social network. In the alumni social network, we investigate the alumni network from a university, which is comprised of 17,381 students graduated from its Computer Science department and all faculty members of University. The network contains 2,113,345 relationships of different types (e.g., colleague, advisor-advisee, classmate, high-school alumni, etc.).

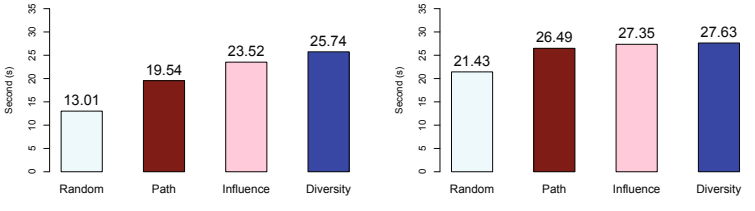
Evaluation Measures. To evaluate the proposed method, we consider two aspects: user's average viewing time and the average number of clicks. User's viewing time stands for how long a user will stay on the returned social graph. Staying for a long time implies that the user may be more interested in the result than that with a shorter time. We also design a user interactive mechanism, which allows the user to expand a person's detailed social information when she/he is interested in knowing more about the person or to remove the node from the returned graph when she/he think the node is irrelevant. For each query, we randomly select one of the proposed three algorithms to generate and return the social graph to the user. We record the user behaviors (viewing time and #clicks) on the returned social graph. We also compare the three algorithms with a baseline algorithm, which randomly selects nodes from the candidate nodes.



(a) #User clicks



(b) Expand/Remove ratio. (Left: Coauthor; Right: Alumni)



(c) Viewing time (Second). (Left: Coauthor; Right: Alumni)

Fig. 2. Performance on the two networks (Coauthor and Alumni)

4.2 Accuracy Performance

As all the comparison methods require the number of users' access and log, we set up the two systems from early 2011. We use the log of four months (March - June, 2011) in the coauthor system (consisting of 57,494 queries) and the log of one month (April, 2011) in the alumni system (consisting of 4,305 queries) to study the performance of different algorithms.Figure 2 shows the results on the coauthor network data and alumni network data.

Effect of User Clicking. Figure 2(a) shows the probability of a user clicking a node in the social graph. Expand indicates that the user clicks to see more detailed person's social circle, while Remove indicates that the user clicks to remove a person from the social graph. We see that all the presented four algorithms attract much higher click ratio

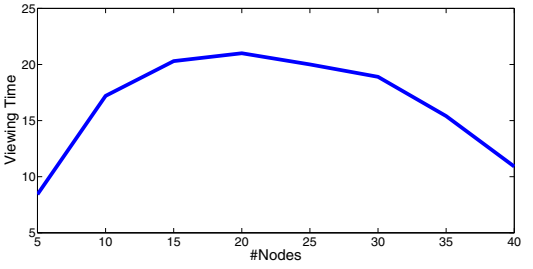


Fig. 3. Viewing time of the number of displayed nodes

than the Random algorithm. An interesting phenomenon is that overall the Path algorithm attracts the largest number of user clicks; however, there are also a large number of users click to remove person nodes from the social graph, which implies that there are not only many "interesting" nodes in social graph returned by the Path algorithm, but also many "irrelevant" nodes. To quantify this, we define another measurement called Expand/Remove ratio as ratio of the number of "Expand" clicks divided by the number of "Remove" clicks. Figure 2(b) shows the result of Expand/Remove ratio by the comparison algorithms. It can be seen that the Diversity algorithm has the largest ratio, while the Random and the Path algorithm have lower ratios.

Effect of User Viewing Time. Figure 2(c) shows the average viewing time of a user on the returned social graph by applying the different algorithms. It can be seen again that the Diversity algorithm results in the longest viewing time, which confirms the findings from Figure 2(b). On average, the presented three algorithms can gain an 73.69%-84.13% increase in terms of the number of (Expand) clicks, and an increase from 34.56%-131.37% in terms of viewing time compared with the baseline (Random) algorithm. In particular, the Diversity algorithm achieves the best performance from the perspective of both Expand/Remove ratio and viewing time.

4.3 Analysis and Discussions

To obtain deeper understanding of the results, we perform the following analysis.

Effect of the Number of Displayed Nodes. We conduct an experiment to see the effect of the number of the displayed nodes. We use the users' average time of display different nodes to overall performance. The curves of coauthor and alumni network look almost the same. As an example, Figure 3 shows the users spend time on different nods. This suggests that about twenty nodes are good display property.

Error Analysis. We conduct an error analysis on the results of our approach. We observe two major types of source of errors.

- Missing data. Sometimes the data is missing because the database does not contain all the coauthor (alumni) relations. For example, there are thousands of papers every year and many different of alumni relations, the database cannot contain all the relations. Thus, the social graph might not also generate the result every time.
- Name ambiguity. In the coauthor network, there might be several persons with the same name. This lead to mistake relationships between persons.

5 Related Work

Social graph is an important problem in social network analysis, Tang et al. [18] study the problem of topic-level social network search, which aims to find who are the most influential users [17] in a network on a specific topic and how the influential users connect with each other. In this section, we review the related work on connectivity subgraphs and diversity.

Connectivity Subgraphs. Social graph search is to find a connectivity subgraph among queried users. Faloutsos et al. [5] also address that problem. The main point of that paper is to develop measures based on electrical-current flows of proximity between nodes of the graph that depend on the global graph structure. And there are many ideas, such as Koren et al. [11] refined the proximity measures using the notion of cycle-free effective conductance. The main difference between our approach and above research is that we define users' influence of each person to others and consider the diversity of the subgraph.

Diversity. Diversity is well-recognized as highly property in many data mining tasks, which is very useful to address uncertainty about the information need given a query. One of the most representative works is on expertise search, such as Agrawal et al. [1] and Gollapudi et al. [6]. There are also some works which have focused on diversity result in recommendation. For example, Ziegler et al. [23]. More recently, Tong et al. propose a new approach for diversity of graph search [21]. The difference of our work from existing lies in that we consider the diversity in the resultant social graphs.

The work is also related to the social relationship mining. For example, Tang et al. [20] propose a learning framework based on partially labeled factor graphs for inferring the types of social relationships in different networks. Tang et al. [16] further study the problem of inferring social ties across heterogeneous networks. However, these methodologies do not consider the network search problem.

6 Conclusions

In this paper, we study a novel problem of instant social graph search, which aims to find a subgraph of representative users to closely connect the queried persons. We formally define this problem and present three algorithms to solve the problem. We have developed two systems to validate the effectiveness and efficiency of the presented algorithms. We have deployed the algorithms in two real systems: an academic mining system and an alumni network system. In terms of both users viewing time and number of clicks, we found that the presented algorithms significantly outperform (+34.56%-+131.37% in terms of viewing time) the baseline method. We also found that the Diversity algorithm can achieve the best performance. The presented algorithms are efficient, and can perform most social graph searches in 2 seconds.

Detecting the personalized social graph represents a new research direction in social network analysis. As further work, it is interesting to study how user's feedback can be used to improve the search performance (e.g., interactive learning).

References

1. Agrawal, R., Gollapudi, S., Halverson, A., Ieong, S.: Diversifying search results. In: WSDM 2009, pp. 5–14 (2009)
2. Aleman-Meza, B., Nagarajan, M., Ramakrishnan, C., Ding, L., Kolari, P., Sheth, A.P., Arpinar, I.B., Joshi, A., Finin, T.: Semantic analytics on social networks: experiences in addressing the problem of conflict of interest detection. In: WWW 2006, pp. 407–416 (2006)
3. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: KDD 2009, pp. 199–207 (2009)
4. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271 (1959)
5. Faloutsos, C., McCurley, K.S., Tomkins, A.: Fast discovery of connection subgraphs. In: KDD 2004, pp. 118–127 (2004)
6. Gollapudi, S., Sharma, A.: An axiomatic approach for result diversification. In: WWW 2009, pp. 381–390. ACM, New York (2009)
7. Hirsch, J.E.: An index to quantify an individual's scientific research output. Proceedings of the National Academy of Sciences 102(46), 16569–16572 (2005)
8. Karypis, G., Kumar, V.: Parallel multilevel k-way partitioning scheme for irregular graphs. In: SC Conference, p. 35 (1996)
9. Kautz, H., Selman, B., Shah, M.: Referral web: Combining social networks and collaborative filtering. Communications of the ACM 40(3), 63–65 (1997)
10. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: KDD 2003, pp. 137–146 (2003)
11. Koren, Y., North, S.C., Volinsky, C.: Measuring and extracting proximity graphs in networks. ACM Trans. Knowl. Discov. Data 1 (December 2007)
12. Landauer, T.K.: Behavioural research methods in human-computer interaction. In: Helander, M., Landauer, T.K., Prabhu, P. (eds.) Handbook of Human-Computer Interaction (1997)
13. Milgram, S.: The small world problem. Psychology Today 2, 60–67 (1967)
14. Ramakrishnan, C., Milnor, W.H., Perry, M., Sheth, A.P.: Discovering informative connection subgraphs in multi-relational graphs. SIGKDD Explor. Newsl. 7, 56–63 (2005)
15. Sozio, M., Gionis, A.: The community-search problem and how to plan a successful cocktail party. In: KDD 2010, pp. 939–948 (2010)
16. Tang, J., Lou, T., Kleinberg, J.: Inferring social ties across heterogenous networks. In: WSDM 2012 (2012)
17. Tang, J., Sun, J., Wang, C., Yang, Z.: Social influence analysis in large-scale networks. In: KDD 2009, pp. 807–816 (2009)
18. Tang, J., Wu, S., Gao, B., Wan, Y.: Topic-level social network search. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2011, pp. 769–772 (2011)
19. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: Extraction and mining of academic social networks. In: KDD 2008, pp. 990–998 (2008)
20. Tang, W., Zhuang, H., Tang, J.: Learning to Infer Social Ties in Large Networks. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011, Part III. LNCS, vol. 6913, pp. 381–397. Springer, Heidelberg (2011)
21. Tong, H., He, J., Wen, Z., Konuru, R., Lin, C.-Y.: Diversified ranking on large graphs: an optimization viewpoint. In: KDD 2011, pp. 1028–1036 (2011)
22. Wang, C., Han, J., Jia, Y., Tang, J., Zhang, D., Yu, Y., Guo, J.: Mining advisor-advisee relationships from research publication networks. In: KDD 2010, pp. 203–212 (2010)
23. Ziegler, C.-N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: WWW 2005, pp. 22–32. ACM, New York (2005)

Peer Matrix Alignment: A New Algorithm

Mohammed Kayed

Faculty of Science, Beni-Suef University, Egypt
mskayed@yahoo.com

Abstract. Web data extraction has been one of the keys for web content mining that tries to understand Web pages and discover valuable information from them. Most of the developed Web data extraction systems have used data (string/tree) alignment techniques. In this paper, we suggest a new algorithm for multiple string (peer matrix) alignment. Each row in the matrix represents one string of characters, where every character (symbol) corresponds to a subtree in the DOM tree of a web page. Two subtrees take the same symbol in the peer matrix if they are similar, where similarity can be measured using either structural, content, or visual information. Our algorithm is not a generalization of 2-strings alignment; it looks at multiple strings at the same time. Also, our algorithm considers the common problems in the field of Web data extraction: missing, multi-valued, multi-ordering, and disjunctive attributes. The experiments show a perfect alignment result with the matrices constructed from the nodes closed to the top (root) and an encourage result for the nodes closed to the leaves of the DOM trees of the test web pages.

Keywords: Text Alignment, Tree Alignment, Web Data Extraction, Information Extraction.

1 Introduction

Deep web presents a huge amount of useful information which is usually formatted for its users. So, extracting relevant data from various sources to be used in web applications becomes not easy. Therefore, the availability of robust, flexible Information Extraction (IE) or Wrapper Induction (WI) systems that transform Web pages into program-friendly structures such as a relational database will become a great necessity. For relevant data extraction, unsupervised IE systems exploit the fact that web pages of the same Web site share the same template since they are encoded in a consistent manner across all the pages; i.e., these pages are generated with a predefined template by plugging data values. Finding such a common template requires as input multiple pages (page-wide IE systems; e.g., RoadRunner [1], EXALG [2], and FiVaTech [3]) or a single page containing multiple records (record-wide IE systems; e.g., IEPAD [4], DeLa [5], and DEPTA [6]).

A crucial step for most web data extraction systems (record/page level systems) is alignment: either string alignment (e.g., IEPAD and RoadRunner) or tree alignment (e.g., DEPTA). Alignment of attribute values in multiple data objects (strings/trees) is a challenging task as these attributes are subject to the following variations [7]:

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 268–279, 2012.

- An attribute may have zero or more values in a data-object. If the attribute has zero value, it is called a missing attribute; if it has more than one value, it is called a multi-valued attribute. A book's author may be an example of multi-valued attribute, whereas a special offer is an example of missing attribute.
- The set of attributes (A_1, A_2, …, A_k) may have multiple ordering, i.e., an attribute A_i may have variant positions in different instances of a data-object. We call this attribute a multi-ordering attribute. For example, a movie site might list the release date before the title for movies prior to 1999, but after the title for recent movies.
- An attribute may have variant formats along with different instances of a data object. If the format of an attribute is not fixed, we might need disjunctive rules to generalize all cases. For example, an e-commerce site might list prices in bold face, except for sale prices which are in red. So, price would be an example of a variant-format (disjunctive) attribute in this site. On the other hand, different attributes in a data-object may have the same format, especially in table presentation, where single <TD> tags are used to present various attributes.
- Most IE systems handle input documents as strings of tokens as they are easier to process than strings of characters. Depending on the used tokenization methods, sometimes an attribute cannot be decomposed into individual tokens. Such an attribute is called an untokenized attribute.

Untokenized attributes can be processed by further processing after the alignment step, while missing, multi-valued, multi-ordering, and disjunctive attributes are handled during the alignment step. The effectiveness of an alignment algorithm relies on its capability to handle such problems. In this paper, we suggest a new alignment algorithm to be used as a part of web data extraction systems. Our algorithm considers the above four mentioned problems. Also, to align multiple data objects, our algorithm processes all objects at the same time. Our algorithm doesn't consider the problem of multiple-objects alignment as a generalization of 2-objects alignment. So, it has a global (better) view for the input objects (strings/trees).

The rest of the paper is organized as follows. Section 2 defines the peer matrix alignment problem. Our proposed alignment algorithm and different examples that we suggest to clarify the algorithm are discussed in sections 3 and 4, respectively. Section 5 describes our experiments. The related works are presented in Section 6. Finally, section 7 concludes our work.

2 Problem Definition

String alignment is simpler than tree alignment. Since web pages used with IE systems are tree structured (DOM trees), tree alignment will become a great necessity. Like FiVaTech, our algorithm considers the tree structure of the objects to be aligned, but we simplify the problem by converting it into string alignment as follows. We consider all of the objects (web pages) to be aligned are tree structured and have the same root p. Our algorithm collects all (first-level) child nodes of p in a matrix M, where each column keeps the child nodes for one root node p. Every node in the

matrix takes a symbol which actually denotes a subtree. All similar subtrees (peer nodes) are denoted by the same symbol. Similarity here can be measured by any suitable technique such as tree-edit distance. The peer matrix shown in Fig. 1b is constructed from the three trees of root *p* in Fig. 1a. The matrix has three columns; each one includes all child nodes of one root *p*, where two similar nodes take the same symbol. Now, the problem is transformed into multiple-string (peer matrix) alignment, where each string corresponds to one column in the matrix. Handling of the problems: missing, multi-valued, multi-ordering, and disjunctive attributes is based on this very important alignment step. The output of this step is an aligned list in which missing, repetitive, disjunctive, and multi-ordering patterns can be identified very easily. For example, as shown in Fig. 1d, the output aligned list has one repetitive pattern (BCD), where the two symbols B and D are optional (missing attributes) in this pattern.

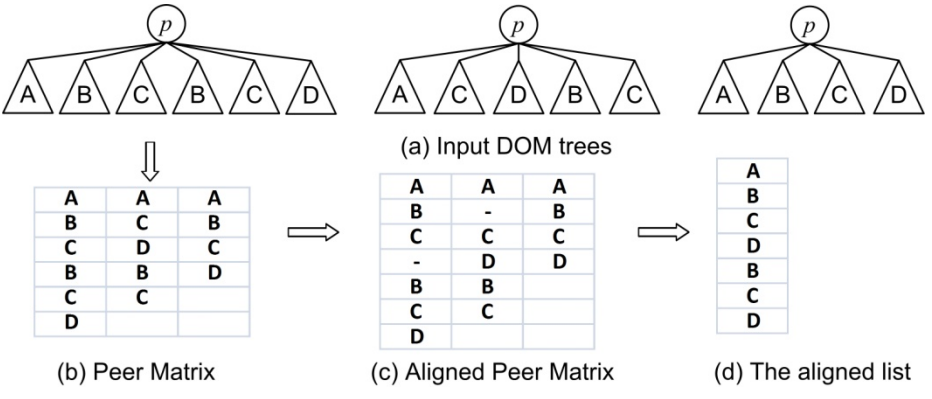


Fig. 1. Peer matrix construction

Definition (Aligned Row): A row in the peer matrix is called an aligned row in two cases. The first case is occurred when all symbols in this row are the same (or disappear in some columns of the row). The second case is occurred when the row has different symbols, but these symbols correspond to leaf nodes (text or <img>) such that each symbol appears only in its residing column (i.e., if a symbol exists in a column c, then all other columns outside the current row in the matrix do not contain this symbol). All of the rows above an aligned row must be also aligned.

Definition (Aligned Peer Matrix): A peer matrix is aligned if all of the rows in the matrix are aligned. Fig. 1c is an example of an aligned peer matrix. As shown in the aligned peer matrix, the symbol '-' refers to a null value which has been added after shifting some symbols down to patch an empty space.

Definition (Aligned List): If M is an aligned peer matrix, so each aligned row in the matrix M is represented by a symbol which is either the same as the symbol in the row (if the row has a same symbol) or an asterisk (if the row has different symbols correspond to leaf nodes). Fig. 1d is the aligned list corresponds to the one in Fig. 1c.

Definition (Peer Matrix Alignment Problem): Given a peer matrix as input, the alignment problem is to modify the content of the matrix by shifting down, swapping, or replacing symbols in the matrix to transform it into an aligned peer matrix.

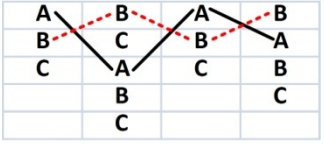


Fig. 2. Examples of regular and irregular zigzag lines L_B and L_A, respectively

3 The Proposed Algorithm

Given a peer matrix as input, we start the matrix row by row (up-down fashion) and align each row using its contents and the contents of the other rows below it, to get an aligned peer matrix. Each row r is aligned by processing of zigzag lines that have been drawn to connect among the symbols in r and other symbols down r in the matrix. Zigzag lines are drawn as follows. For each symbol s in r, we draw a sequence of lines (a zigzag line) L_s that connect among all first occurrences of the symbol s in each column of the current row r or the rows below r. The zigzag line L_s of the symbol s at row r passes through different occurrences of s in different columns in r or below r with at most one occurrence of s in each column. Each occurrence of s is connected by a line with the first occurrence in the right/left hand side column. If the symbol does not appear in the right/left column, the line will pass to connect the first occurrence in the next column, and so on. All occurrences of s in the row r belong to the same zigzag L_s. If the symbol does not appear again in the row or below the row, we call it a *non-zigzag symbol*. This means, a non-zigzag symbol is not belonging to any zigzag line in the row. Fig. 2 shows two zigzag lines L_A and L_B at the first row. By drawing zigzag lines for the symbols in the row r, the challenge here is how we can process these zigzags to align the row r. Although, zigzag lines are different for different columns order in the matrix, our algorithm solves the problem in general and covers all suitable cases. Before we go further in this section to discuss our proposed alignment algorithm, we give some definitions that are important to the algorithm.

Let l is a line in a zigzag L_s that connects two occurrences of s at the two locations (r_1,c_1) and (r_2,c_2) in the matrix, we define the vertical span of l as $(r_2 - r_1+1)$ and the horizontal span as $(c_2 - c_1+1)$. Also, we call a zigzag L_s horizontally covers the peer matrix if it is started at the first column and terminated at the last column of the matrix. The vertical spans of the three lines of L_A in Fig 2 are 3, 3, and 2, respectively. Also, the two zigzag lines L_A and L_B in Fig 2 horizontally cover the peer matrix.

Definition (A Repetitive Pattern): A pattern P (a sequence of one or more consecutive symbols in a column) is called repetitive if it has more than one occurrence in some column of the matrix. If P appears at most once in each column of the matrix, we call it a free (non-repetitive) pattern.

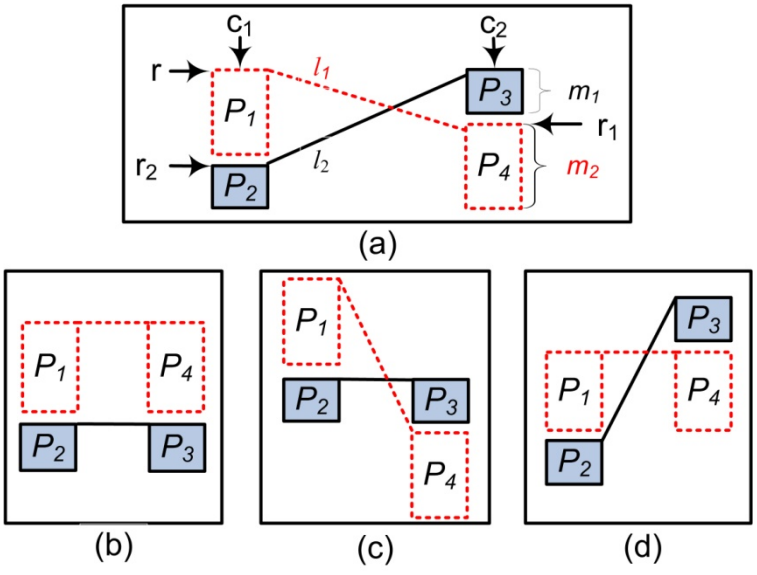


Fig. 3. A left-most cross (a) and three alignment solutions (b), (c), and (d)

Definition (Symbol Span): The span of a symbol *s* is defined as the maximum number of different symbols (without repetition) between any two consecutive occurrences of *s* in each column plus one; i.e., span of *s* represents the maximum possible cycle length of *s*. If *s* is a non-repetitive symbol, then its span value will be 0. For example, the span of the symbol B in the matrix in Fig. 2 is 3 (there are two different symbols, C and A, between the two occurrences in the second column).

In our proposed algorithm, we use the calculated span value for a symbol *s* to control (restrict) the process of shifting down *s* in the matrix. Shifting down the symbol *s* must not violate the calculated symbol span value. This means, if *s* occurred only once in a column, it can be shifted down to anywhere in the column. But, the symbol *s* which is occurred at row *r* and column *c* cannot be shifted down if it appears up in *c* at a row r′ (i.e., r > r′) such that r-r′ ≥ span(*s*). For example, the symbol B in the fourth row and the second column of the matrix in Fig. 2 cannot be shifted down because its span is 3 and it appears up at the first row of the column.

Definition (Regular Zigzag): a zigzag L_s is *regular* if the vertical spans of all of its lines are equal, and it horizontally covers the matrix. Otherwise it is called *irregular*. Fig 2 shows examples of regular and irregular zigzag lines L_B and L_A, respectively.

Definition (A Top Horizontal Zigzag): a zigzag L_s is called horizontal if all of its lines are horizontal. If there is only one horizontal zigzag in a row, we call it a top horizontal. If there is more than one horizontal zigzag line, we call the one with a maximum number of lines a top horizontal zigzag.

Definition (Left-most Cross): Let l_1 is a line ((r,c_1), (r_1,c_2)) which belongs to a zigzag L_a in a row *r* as shown in Fig. 3(a). We call the cross between the line l_2 ((r,c_2),

(r_2,c_1)) in a zigzag L_b and l_1 a left-most cross if the vertical distance between r_2 and r (m_2= r_2-r) is a minimum. As shown in Fig. 3(a), the cross makes four patterns in the peer matrix: P_2 and P_3 are the two patterns of lengths m_1 under the two ending points of l_2, while P_1 and P_4 are the two patterns of lengths m_2 under l_1.

To align the row r which has a left-most cross as in Fig.3 (a), we have three possible different alignment cases. The first case, case I, is occurred if either P_1 or P_3 is optional (missing pattern in c_2 or c_1, respectively, as shown in Figures 3(c) and (d)). The second case, case II, is occurred when the two patterns P_1 (P_3) and P_2 (P_4) are multi-ordering in the column c_1 (c_2), respectively, as shown in Fig. 3(b). Finally, the third case, case III, is occurred when the two patterns P_1 and P_3 have disjunctive attributes (i.e., same data presented in different formats). Our algorithm works as follows to distinguish between these three cases I-III.

If either $P_1 \neq P_4$ or $P_2 \neq P_3$, we align the row as case I (P_1 or P_3 is optional). Particularly, if $P_1 = P_4$ (while $P_2 \neq P_3$), we identify P_3 as optional (i.e., P_1 is shifted down in the column c_1 a distance m_1). If $P_2 = P_3$ (while $P_1 \neq P_4$), we identify P_1 as optional (i.e., P_3 is shifted down in the column c_2 a distance m_2). Finally, if both $P_1 \neq P_4$ and $P_2 \neq P_3$, we shift down either P_1 or P_3 based on some criteria as we will discuss later.

If both $P_1 = P_4$ and $P_2 = P_3$, the algorithm deals with the problem as follows. As in Fig. 3 (c) and (d), to identify the pattern P_1 (P_3) as optional in c_2 (c_1), it is necessary (but not sufficient) that P_1 (P_3) is repetitive in the matrix, respectively. So, if both P_1 and P_3 are non-repetitive patterns, we identify P_1 (P_3) and P_2 (P_4) as multiple-ordering patterns (case II) in the column c_1 (c_2), respectively. The challenge here is occurred when either P_1 or P_3 is repetitive. Experimentally, we have observed that case I (either P_1 or P_3 is optional) is mostly the correct alignment choice. An exception is occurred when P_1 and P_3 have disjunctive patterns (the row is aligned using case III).

Figure 4 shows an example of case III (disjunctive attributes) when a search engine web site presents a list of resulted web pages as links (<A> tag), except the current page is presented using <STRONG> tag (i.e., the two tags <A> and <STRONG> are disjunctive). So, the two cases I and III are possible when either P_1 or P_3 is repetitive. We handle this problem and decide the correct alignment case based on the following assumption. The two patterns a_1 and a_2 are disjunctive, if one of the two patterns (say a_1) appears randomly among different consecutive occurrences of the other one (a_2) in each column of the matrix. So, we assume that, there is a left-most cross in two columns c_1 and c_2 such that one of the two patterns P_1 or P_3 is a sequence of the repetitive pattern a_2, while the other one is a_1. If so, we replace all occurrences (in the matrix) of a_1 by a_2 and mark a_1 as disjunctive with a_2. In the example shown in Fig. 4, P_3 is a sequence of the repetitive pattern a_2 (<A>), and P_1 = <STRONG>. So, we identify a_1 (<STRONG>) and a_2 (<A>) as disjunctive patterns. Our proposed algorithm which handles all of these cases and others to align the row r is shown in Fig. 5.

As shown in Fig. 5, to align the row r in M, the algorithm recursively tries to align r and stops at three base cases: first (lines 2-3), when r is an aligned row as defined before, second (lines 4-6), when r only has disjunctive attributes, and third (lines 7-11) when r has a top-horizontal zigzag. In the first case, the algorithm returns a symbol s when r has either a horizontal zigzag L_s or a non-zigzag symbol s. But, if r has leaf nodes (img/text), the algorithm returns "*". In the second case, the algorithm checks if r only has disjunctive attributes by using DisjunctiveAttributes(r, M). The function returns true if both of the following two conditions are satisfied:

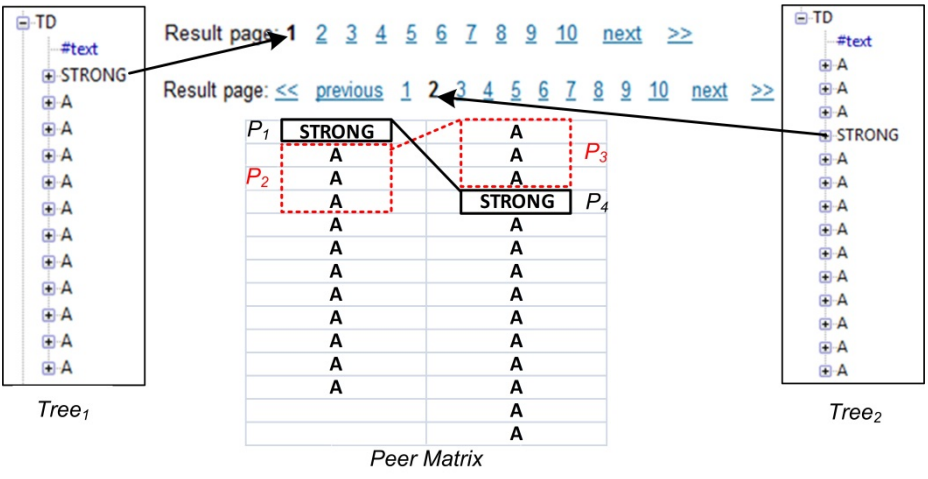


Fig. 4. An example of disjunctive attributes in the matrix constructed from $Tree_1$ and $Tree_2$

```
Algorithm AlignRow(r, M)                    // To align the row r of the peer matrix M.

1.  for each different symbol s_i (i=1, 2, …, c) in the row r, draw a zigzag line L_si;
2.  if (r is an aligned row)                // r is aligned if it has only one horizontal zigzag L_s, r has only one
3.      return s;                           // non-zigzag symbol s, or r has variant leaf nodes (i.e., s="*").
4.  else if (DisjunctiveAttributes(r, M) ))                                   // Case i
5.      s = s_1 || s_2 || …|| s_c ;                    // Disjunctive Attributes
6.      return s;
7.  else if (there is a top horizontal zigzag line L_s)                       // Case ii
8.      shiftNonZigzagSymbols(r, M);                   // Missing attributes
9.      shiftOtherHorizontalZigzags(r, M);             // Missing attributes
10.     stretchZigzag(L_k) for each zigzag L_k (k≠s) in r;   // Missing attributes
11.     return s;
12. else if (there is only ONE non-horizontal regular/irregular zigzag L_s)   // Case iii
13.      stretchZigzag(L_s);                           // Missing attributes
14.      AlignRow(r, M);
15. else                                    // there is a left-most cross.
16.     let m_1, m_2 and P_1, P_2, P_3, and P_4 are defined for L_a and L_b as shown in Fig. 3(a);
17.     if ( either (P_2≠ P_3) or (P_1≠P_4) )          // Missing attributes     Case I
18.         stretchZigzag( selectAZigzag(L_a, L_b, r, M) );
19.         AlignedRow(r, M);
20.     else if (both P_1 and P_3 are non-repetitive)  // Multiple-ordering attributes   Case II
21.         exchange the two patterns P_3 (P_1) and P_4 (P_2) in the column c_2 (c_1), respectively;
22.         AlignRow(r, M);
23.     else if (one pattern (P_1 or P_3) is a seq. of a repetitive pattern a_2, the other equals a_1)   // Case III
24.         replace all occurrences of a_1 by a_2, mark a_1 as disjunctive with a_2;  // Disjunctive patterns
25.         AlignRow(r, M);
26.     else                                           // Missing attributes     Case I
27.         stretchZigzag( selectAZigzag (L_a, L_b, r, M) );
28.         AlignRow(r, M);
29.     endif
30. endif
```

Fig. 5. Our proposed algorithm to align a row r in a peer matrix M

- The row r has a sequence of different symbols s_1, s_2, …, s_k; $k > 1$, where each symbol s_i is either a non-zigzag symbol or belonging to a horizontal zigzag line.
- For each different symbol s_i in r, there is a zigzag line L_s that connects among different occurrences of some symbol s below s_i.

The first condition makes sure the symbols in r have no occurrences below r, while the second condition gives a guarantee that there exists a zigzag which prevents any of these symbols to be shifted down. The third stop case is occurred when the row r has a combination of horizontal zigzag lines (one of them is a top-horizontal), non-zigzag symbols, and other regular/irregular zigzag lines. If so, the algorithm only keeps the top-horizontal zigzag in the row r and considers all others as missing attributes. Therefore, it shifts all non-zigzag symbols (line 8) and all horizontal zigzag lines (line 9) down a distance 1, and stretches all remaining regular/irregular zigzag lines (line 10) using the function stretchZigzag(L_s). The function stretchZigzag works as follows. If L_s is regular (i.e., L_s connects among different occurrences of s in the two rows r and r'; $r<r'$), it shifts all occurrences of s in r downward a distance r'-r in the matrix M and patch empty spaces with a null value. If L_s is irregular (i.e., L_s connects among different occurrences of s in r and other different rows r_1, …, r_k below r), the function shifts each occurrence of s at each row r_i above r´ downward a distance r'-r in the matrix M and patch empty spaces with a null value, where the row r´∈ $\{r_1, \ldots, r_k\}$ satisfies that r'-r is min(r_1-r, …, r_k-r).

When the row has one regular/irregular zigzag line, the algorithm identifies it as missing attribute (lines 12-14) and uses the function stretchZigzag to stretch it. Finally, if the row has a left-most cross (lines 16-29), the algorithm handles it as we discussed above. The function selectAZigzag(L_a, L_b, r, M) returns one of the two zigzag lines L_a or L_b to be stretched based on either of the following three ordered criteria: First, a zigzag that has a line of non-zero minimum vertical span is returned. Second, the one with a maximum number of horizontal lines in the row r of the matrix M is returned. Third, the algorithm returns the right-most one.

For an $n \times m$ matrix M, the running time to draw zigzag lines for each row is $O(n \times m)$. Also, the running time to check whether some pattern is repetitive or not is $O(n \times m)$. As a preprocessing step, for each row, the running time for calculating symbols scan values is $O(n \times m)$. Therefore, the running time for each call of the recursive algorithm to align a row r in the matrix M is $O(n^2 \times m^2)$. Experimentally, a row is aligned after 2-3 calls.

4 Examples

The two matrices in Fig. 6 give two examples of disjunctive attributes. To align the first row of the matrix in Fig. 6(a), the row has one horizontal zigzag L_A and one non-zigzag symbol F, and at the same time there is a zigzag L_C (in the third row) which has occurrences of C below A and F. So, the algorithm identifies A and F as disjunctive attributes (returns s=A||F). To align the first row of the matrix in Fig. 6(b), the row has two horizontal zigzag lines L_A and L_F, and at the same time there is a zigzag L_C which has occurrences of C below both A and F. So, our algorithm also identifies A and F as two disjunctive attributes, and then returns s=A||F.

Fig. 7 gives two examples of a top horizontal zigzag (case ii in Fig. 5). To align the first row of the first matrix (Fig. 7(a)), the row has two horizontal zigzag lines L_A (the top one because it has the maximum number of lines: 2) and L_F. However, there is no zigzag lines that have occurrences below A and F at the same time (i.e., A and F are not identified as disjunctive attributes). So, the algorithm shifts down L_F to the next

row in the matrix. For the matrix in Fig. 7(b), the first row has one horizontal zigzag L_A (the top one) and one regular zigzag L_B. So, the regular zigzag L_B is stretched at the second row (i.e., all occurrences of B in the first row will be shifted down a distance 1 and patch empty spaces with a null value).

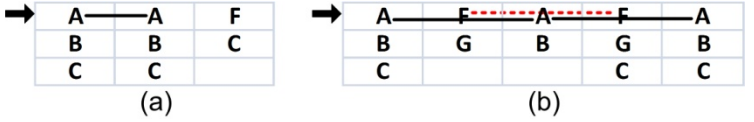


Fig. 6. Two examples of disjunctive attributes, case i

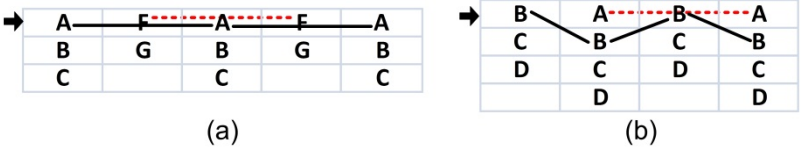


Fig. 7. A row has a top horizontal zigzag, case ii

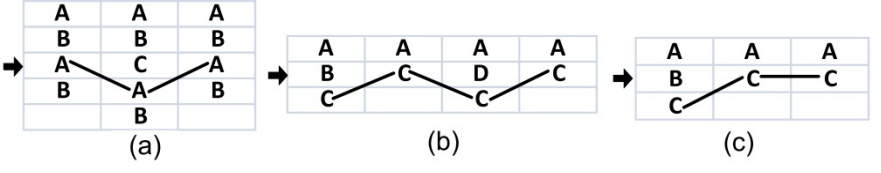


Fig. 8. A row has either one regular zigzag (a) and (b), or one irregular zigzag (c); case iii

Fig. 8 gives three examples when the row has only one non-horizontal regular zigzag (case iii). To align the third (second) row of the matrix in Fig. 8(a) (Fig.8(b)), the algorithm (line 13 in Fig. 5) stretches down L_A (L_C) at the fourth (third) row, respectively. Fig. 8(c) gives an example when the row has only one irregular zigzag (L_C). To align this row, L_C is stretched down (line 13 in Fig. 5) at the third row.

Fig. 9 gives two examples of missing attributes (case I in Fig. 5), when there is a combination of regular and irregular zigzag lines that form one or more crosses. The first step here is to identify the left-most cross and calculate m_1, m_2, and P_1- P_4 as shown in Fig. 3(a). Fig. 9 (a) presents an example of missing attributes when either $P_1 \neq P_4$ or $P_2 \neq P_3$, while Fig. 9 (b) presents another example when both $P_1 = P_4$ and $P_2 = P_3$, either P_1 or P_3 is repetitive, and none of the two patterns P_1 and P_2 is a sequence of some repetitive pattern a_2. In the two examples, the function selectAZigzag in Fig. 5 is used to select one of the two zigzag lines (that make a left-most cross) to be stretched down. For the first matrix in Fig. 9, the function selects L_C to be stretched at the third row because it has a line of the minimum vertical span (2). Also, it selects L_A to be stretched down for the same reason in the second matrix.

Fig. 10 discusses the case of multiple-order attributes, where $P_1 = P_4$="ABC" and $P_2 = P_3$="FGH", but both of the two patterns P_1 and P_3 are non-repetitive. So, to align the first row of the matrix, our algorithm exchanges the two patterns P_2 and P_3 in the third column because they are multiple-ordering patterns. We shall not give here any example of case III, because we already presented one in the previous section.

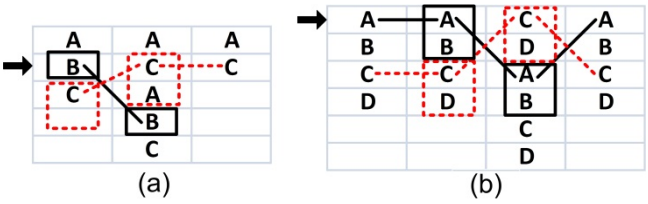


Fig. 9. Two examples of missing attributes, case I

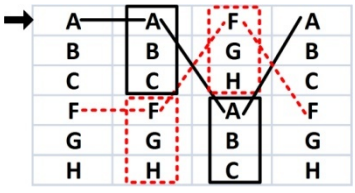


Fig. 10. Multiple-ordering attributes, case II

5 Experiments

We measure the performance of the proposed algorithm by collecting 300 peer matrices taken from a data set of 10 web sites (selected from the manually labeled Testbed for Information Extraction from Deep Web TBDW [8] Version 1.02) as follows. For each web site, we randomly select 30 matrices: 10 matrices from top levels (closed to the root), 10 matrices from the levels closed to the leaves, and 10 matrices from the whole DOM tree. The performance of the algorithm is measured by calculating recall and precision for each selected matrix as follows. Precision is the proportion of symbols predicted by the algorithm as missing, disjunctive, or multiple-ordering that are targets (correctly identified). Recall is the proportion of missing, disjunctive, or multiple-ordering symbols that are predicted by the algorithm. The terms true positives (T_p), true negatives (T_n), false positives (F_p) and false negatives (F_n) compare the predicted class of matrix symbols with the actual class. T_p is the number of missing, disjunctive, or multiple-ordering symbols that are correctly identified by the algorithm. F_n is the number of missing, disjunctive, or multiple-ordering symbols that cannot be identified by the algorithm. F_p is the number of missing, disjunctive, or multiple-ordering symbols that are incorrectly identified by the algorithm. Finally, T_n is the number of symbols that are correctly not identified by the algorithm as missing, disjunctive, or multiple-ordering. Formally, we define recall and precision as follows:

$$Recall = \frac{T_p}{T_p + F_n}; \qquad Precision = \frac{T_p}{T_p + F_p}$$

The performance of the algorithm with the 30 web sites is shown in Table 1. For each web site, the average of the calculated recall and precision for the selected 10 matrices constructed closed to the root (closed to the leaves and from the whole DOM tree) is shown in column 2-3 (4-5 and 6-7, respectively). As shown in the table, the algorithm performs perfectly with matrices near to the root of a DOM tree (columns 2-3), and

gives a good result near to leaves (columns 4-5) as the matrices near to the leaves are complicated and contain many missing, disjunctive, and multi-ordering symbols. The perfect results for the matrices near to the root was not a surprise because many of the matrices are already/easy to be aligned. In general, the results are encouraged for the whole DOM tree (columns 6-7).

Table 1. The performance of the algorithm with a data set of 10 web sites

Site	Closed to root		Closed to leaves		Whole DOM tree	
	Recall	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**
Allnealthnet	1.00	1.00	0.95	0.92	0.99	0.95
G. Unlimited	1.00	1.00	1.00	0.99	1.00	0.97
IUMA	1.00	1.00	0.93	0.92	0.95	0.92
Picsearch	1.00	1.00	1.00	1.00	1.00	1.00
Sun-Sentinel	1.00	1.00	0.98	0.96	1.00	0.97
amazon.co.uk	1.00	1.00	0.91	0.90	0.91	0.91
Amazon	1.00	1.00	0.99	0.95	0.98	0.96
Gene surname	1.00	1.00	0.95	0.95	0.95	0.97
HomePopular	1.00	1.00	0.99	0.97	1.00	0.98
NAMI	1.00	1.00	0.97	0.96	0.98	0.97
Average	**1.00**	**1.00**	**0.97**	**0.95**	**0.98**	**0.96**

6 Related Works

IEPAD [4] and OLERA [9] generalize extraction patterns from unlabeled Web pages. Repetitive patterns in IEPAD are discovered using the binary suffix tree PAT tree. PAT trees compute only exact match patterns, templates with exceptions cannot be discovered through PAT trees. Patterns with inexact or approximate matching are discovered using multiple string alignment technique. IEPAD applies center star algorithm to align multiple strings. In OLERA [9], user marks a record to be extracted to discover other similar records and generalize them using multiple string alignment. OLERA handles the problem in IEPAD when several alignments exist by proposing a matching function to compare the primitive data for text tokens.

RoadRunner [1] considers the site generation process as encoding of the original database content into strings of HTML code. The system uses the ACME matching (alignment) technique to compare HTML pages of the same class and generate a wrapper based on their similarities and differences.

DEPTA [6] discovers repetitive patterns by comparing adjacent substrings with starting tags having the same parent in the HTML tag tree. The recognition of data items or attributes in a record is accomplished by partial tree alignment. The algorithm first chooses the record tree with the largest number of data items as center and then matches other record trees to the center tree. ViPER [10] assumes that repetitive patterns have variant lengths rather than they are of fixed length as in Depta. ViPER applies a tandem repeats algorithm before computing the edit-distance to handle missing and multiple values data. It applies a data alignment technique that is based on

global matching and text content information. The alignment method uses a divide-and-conquer fashion to reduce the multiple-alignment problem.

Finally, FiVaTech [3] conducts four steps: peer node recognition, matrix alignment, pattern mining, and optional node detection in turn. In the matrix alignment step, the system handles the two problems of disjunctive and multiple-ordering attributes as a case of missing attributes.

7 Conclusions

In this paper, we proposed a new algorithm for multiple string (peer matrix) alignment. Our algorithm looked at all of the multiple strings at the same time, so it has a global view for the inputted strings. Also, our algorithm considered the common problems in the field of web data extraction: missing, multi-valued, multi-order, and disjunctive attributes. To align a row in the peer matrix, our algorithm drew some virtual zigzag lines for each symbol in the row, and then tried to stretch/shift some lines to accomplish the task.

References

1. Crescenzi, V., Mecca, G., Merialdo, P.: Knowledge and Data Engineerings. In: Proc. Int'l Conf. Very Large Databases (VLDB), pp. 109–118 (2001)
2. Arasu, A., Garcia-Molina, H.: Extracting Structured Data from Web Pages. In: Proc. ACM SIGMOD, pp. 337–348 (2003)
3. Kayed, M., Chang, C.-H.: Page-level web data extraction from template pages. IEEE Trans. Know. and Data Eng. 22(2), 249–263 (2010)
4. Chang, C.-H., Lui, S.-C.: IEPAD: Information Extraction Based on Pattern Discovery. In: Proc. Int'l Conf. World Wide Web (WWW-10), pp. 223–231 (2001)
5. Wang, J., Lochovsky, F.H.: Data Extraction and Label Assignment for Web Databases. In: Proc. Int'l Conf. World Wide Web (WWW-12), pp. 187–196 (2003)
6. Zhai, Y., Liu, B.: Web Data Extraction Based on Partial Tree Alignment. In: Proc. Int'l Conf. World Wide Web (WWW-14), pp. 76–85 (2005)
7. Chang, C.-H., Kayed, M., Girgis, M., Shaalan, K.: Survey of Web Information Extraction Systems. IEEE Trans. Know. and Data Eng. 18(10), 1411–1428 (2006)
8. Yamada, Y., Craswell, N., Nakatoh, T., Hirokawa, S.: Testbed for Information Extraction from Deep Web. In: Proc. WWW-13, pp. 346–347 (2004)
9. Chang, C.-H., Kuo, S.-C.: OLERA: A semi-supervised approach for Web data extraction with visual support. IEEE Intelligent Systems 19(6), 56–64 (2004)
10. Simon, K., Lausen, G.: ViPER: Augmenting Automatic Information Extraction with Visual Perceptions. In: Proc. CIKM (2005)

Domain Transfer Dimensionality Reduction via Discriminant Kernel Learning

Ming Zeng and Jiangtao Ren

Sun Yat-Sen University, Guangzhou, 510006, China
mingtsang.zm@gmail.com,
issrjt@mail.sysu.edu.cn

Abstract. Kernel discriminant analysis (KDA) is a popular technique for discriminative dimensionality reduction in data analysis. But, when a limited number of labeled data is available, it is often hard to extract the required low dimensional representation from a high dimensional feature space. Thus, one expects to improve the performance with the labeled data in other domains. In this paper, we propose a method, referred to as the domain transfer discriminant kernel learning (DTDKL), to find the optimal kernel by using the other labeled data from out-of-domain distribution to carry out discriminant dimensionality reduction. Our method learns a kernel function and discriminative projection by maximizing the Fisher discriminant distance and minimizing the mismatch between the in-domain and out-of-domain distributions simultaneously, by which we may get a better feature space for discriminative dimensionality reduction with cross-domain.

Keywords: Discriminant Kernel Learning, Dimensionality Reduction, Transfer Learning.

1 Introduction

In many real-world applications, such as image processing, computational biology and natural language processing, the dimensionality of data is usually very high. Due to the complexity and noise of high-dimensional data, the effectiveness of regression or classification is limited. This can be improved via dimensionality reduction which finds a compact representation of the data for classification.

A more popular technique for dimensionality reduction is discriminant analysis. To handle nonlinear problems, the kernel discriminant analysis (KDA) is proposed in [1], which computes the discriminative projection from the data set that is mapped nonlinearly into the reproducing kernel Hilbert space (RKHS). We observe that the kernel is chosen before learning in the KDA method. However, the kernel-based learning methods are desirable when integrating the tuning of kernel into the learning space.

In addition, the discriminant multiple kernel learning methods require a plenty of labeled samples to discriminate the unlabeled data from each class. In real-world applications, it is usually costly or even impossible to get such a huge

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 280–291, 2012.

number of labeled samples from the same distribution. When this situation occurs, the performance of discriminant kernel learning methods is poor. Then one expects to carry out discriminant analysis with the help of other related labeled data from other domains. This brings out the cross-domain problem since the existing discriminant kernel learning makes the assumption that the training data and the test data are independent and identical. To resolve this problem, cross transfer learning is proposed, whose aim is to improve learning in the in-domain by porting the labeled sample from out-of-domain to that from in-domain to carry out dimensionality reduction. Several works have been done by combining unsupervised dimensionality with clustering, such as transferred dimensionality analysis(TDA) [2], which intends to select the most discriminative subspace and clustering at the same time. Maximum mean discrepancy embedding(MMDE)[3] tries to find a subspace where training and test samples distribute similarly to solve the sample selection bias problem in an unsupervised way. S.Si et al.[4] proposed using evolutionary cross-domain discriminative Hessian eigenmaps by minimizing the quadratic distance between the distribution of the training set and that of the test set. However, it could not solve non-linear problems.

In this paper, we develop a new dimensionality reduction method, called domain transfer discriminant kernel learning method (DTDKL), which transfers the knowledge from labeled data in out-of-domain to the in-domain by explicitly carrying out kernel discriminant learning and transfer leaning in a coherent way. More specifically, DTDKL tries to find a projection to maximize the Fisher discriminant ratio in the optimal feature space and minimize the maximum mean discrepancy (MMD) of the different distributions simultaneously. In fact, DTDKL provides a method to learn an optimal kernel function and discriminant projection at the same time.
The key contributions of the paper can be highlighted as follows:

- To the best of our knowledge, DTDKL is the first semi-supervised cross-domain discriminant kernel learning method. In contrast to the prior discriminant kernel learning method, DTDKL does not assume that the training and test data are drawn from the same distribution. Moreover, a novel dimensionality reduction method with cross-domain is proposed, whose objects are to maximize the Fisher discriminant ratio while minimizing the maximum mean discrepancy of different distributions.
- By comparing the state-of-the-art dimensionality reduction methods, DTDKL performs better in the dataset of SyskillWebert, Reuters-21578 and 20-Newsgroup ensuring promising performance in real applications.

The rest of this paper is organized as follows: Section 2 presents the related works and preliminaries of DTDKL; Section 3 proposes DTDKL method by embedding maximum mean discrepancy (MMD) into discriminant analysis to tackle the cross-domain problem; Section 4 presents our experimental results to demonstrate its applications. Finally, we conclude the study in Section 5.

2 Brief Review of Prior Work

2.1 Discriminant Multiple Kernel Learning

Dimensionality reduction has always attracted amount of attention. Various methods have been proposed in a recent survey [5] to solve this problem. The canonical dimensionality reduction algorithm is linear discriminant analysis (LDA) [6], which is finding the most discriminative subspace for different classes in the original space. And with the development of kernel-based methods, kernel discriminant analysis has received a lot of interest for nonlinear problems. The KDA algorithm finds the direction in a feature space, defined by a kernel function, onto which the projections of different classes are well separated [1,7]. Note that the kernel function plays a crucial role in kernel methods, and Lanckriet et al. [8] pioneered the work of multiple kernel learning (MKL) in which the optimal kernel is obtained as a linear combination of pre-determined kernel matrices. Based on ideas of MKL, the kernel-based learning method for discriminant analysis was reformulated as semi-definite programming (SDP) in Kim et al. [9]. Ye et al. [10] improved the efficiency of the problem and extended naturally to the multi-class setting by casting the SDP formulation in quadratically constrained quadratic programming (QCQP) and semi-infinite linear programming (SILP).

2.2 Transfer Learning and Maximum Mean Discrepancy Formulation

Semi-supervised learning aims to make use of unlabeled data in the process of supervised learning and it has also been widely used in many areas related to transfer learning. One of the typical branches is to find criteria to estimate the distance between different distributions. A well-known example is Kullback-Leibler (K-L) divergence. Many criteria are parametric for the reason that an intermediate density estimate is usually required. To avoid parametric estimation, some nonparametric methods are proposed to evaluate the distance between the different distributions of data sets. Maximum Mean Discrepancy (MMD) is a effective nonparametric criterion for comparing distributions based on RKHS [11]. Suppose $\mathcal{F}$ be a class of functions $f : \mathcal{X} \to \mathbb{R}$, and $X = (x_1, \ldots, x_{n_1})$, $Y = (y_1, \ldots, y_{n_2})$ be random variable sets drawn from distributions $\mathcal{P}$ and $\mathcal{Q}$, respectively. The maximum mean discrepancy and its empirical estimate is as follows:

$$\mathrm{MMD}[\mathcal{F}, X, Y] := \sup_{f \in \mathcal{F}} \left(\frac{1}{n_1} \sum_{i=1}^{n_1} f(x_i) - \frac{1}{n_2} \sum_{i=1}^{n_2} f(y_i) \right) \tag{1}$$

The function space $\mathcal{F}$ could be replaced by $\mathcal{H}$ which is a universal RKHS. By the fact that in RKHS, $f(x)$ can be expressed as an inner product via $f(x) = \langle \varphi(x), f \rangle_{\mathcal{H}}$, where $\varphi(x) : \mathcal{X} \to \mathcal{H}$, then one may rewrite MMD as follows:

$$\mathrm{MMD} = \left\| \frac{1}{n_1} \sum_{i=1}^{n_1} \varphi(x_i) - \frac{1}{n_2} \sum_{i=1}^{n_2} \varphi(y_i) \right\|_{\mathcal{H}} = \|\mu_1 - \mu_2\|_{\mathcal{H}}$$

In terms of the MMD theory [11], the distance between distributions of two samples is equivalent to the distance between the means of the two samples mapped into a RKHS.

3 Semi-supervised Discriminant Analysis in Cross-Domain

Let $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \{-1, +1\}$ denote the input space and the output space, respectively. Let $\mathcal{D}^{out} = \{(x_i^{out}, y_i^{out}) \in \mathcal{X} \times \mathcal{Y} : 1 \leq i \leq n^{out}\}$ be the set of out-of-domain data samples with $n^{out} = |\mathcal{D}^{out}|$, and $\mathcal{D}^{in} = \mathcal{D}_l^{in} \cup \mathcal{D}_u^{in}$ be the set of in-domain data samples where $\mathcal{D}_l^{in} = \{(x_i^{in}, y_i^{in}) \in \mathcal{X} \times \mathcal{Y} : 1 \leq i \leq n_l^{in}\}$ with $n_l^{in} = |\mathcal{D}_l^{in}|$, and $\mathcal{D}_u^{in} = \{x_i^{in} \in \mathcal{X} : n_{l+1}^{in} \leq i \leq n^{in}\}$ with $n_u^{in} = |\mathcal{D}_u^{in}|$. Typically, $n_l^{in} \ll n_u^{in}$. Let $\mathcal{P}$ and $\mathcal{Q}$ be the marginal distribution of $\mathcal{D}^{in}$ and $\mathcal{D}^{out}$, respectively. We assume that the n^{out} out-of-domain samples and the n^{in} in-domain samples are drawn independently and identically from a fixed but unknown underlying probability distribution $\mathcal{P}$ and $\mathcal{Q}$, respectively. Our task is to predict the labels $y_{n_l+1}^{in}, \ldots, y_n^{in}$, which corresponds to the inputs $x_{n_l+1}^{in}, \ldots, x_n^{in}$ in the in-domain data set.

3.1 Standard Discriminant Kernel Learning Analysis

The standard kernel discriminant analysis learns the kernel and the direction from the labeled samples in $\mathcal{D}_l^{in}$ in order to project the unlabeled samples in $\mathcal{D}_u^{in}$. Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function. Then, Mercer's Theorem [12] tells us the kernel function implicitly maps the input space $\mathcal{X}$ to a high-dimensional (possibly infinite) Hilbert space $\mathcal{H}$ equipped with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ through a map $\varphi : K \to \mathcal{H}$:

$$K(x, z) = \langle \varphi(x), \varphi(z) \rangle_{\mathcal{H}}, \quad \forall x, z \in \mathcal{X}.$$

This space is called the feature space, and the mapping is called the feature mapping. They depend on the kernel function K and will be denoted as φ_K and $\mathcal{H}_K$.

Let x_+ and x_- denote the collection of data points from positive and negative classes, respectively. Then the total number of data points in the training set $\mathcal{D}_l^{in}$ is $n_l = n_+ + n_-$. The standard kernel discriminant analysis [9] learns the kernel K and direction $w \in \mathcal{H}_k$ via the optimization problem

$$\begin{aligned} \max_{w,K} \; & F(w, K) = \frac{(w^T(\mu_K^+ - \mu_K^-))^2}{w^T(\Sigma_K^+ + \Sigma_K^- + \lambda I)w} \\ \text{s.t.} \quad & K = \sum_{i=1}^{p} \theta_i K_i, \; \mathbf{1}^T\theta = 1, \; \theta \succeq 0, \end{aligned} \tag{2}$$

where $K_1, \ldots, K_p$ be the given p based kernels, $\theta \succeq 0$ means its elements θ_i are nonnegative, $\lambda > 0$ is a regularization parameter, I is the identity operator in $\mathcal{H}_K$, μ_K^+ and μ_K^- are the sample means

$$\mu_K^+ = \frac{1}{n_+} \sum_{i=1}^{n_+} \varphi_K(x_i), \quad \mu_K^- = \frac{1}{n_-} \sum_{i=n_++1}^{n_l} \varphi_K(x_i),$$

and Σ_K^+ and Σ_K^- are the sample covariances

$$\Sigma_K^+ = \frac{1}{n_+}\sum_{i=1}^{n_+}(\varphi_K(x_i) - \mu_K^+)(\varphi_K(x_i) - \mu_K^+)^T,$$
$$\Sigma_K^- = \frac{1}{n_-}\sum_{i=n_++1}^{n_l}(\varphi_K(x_i) - \mu_K^-)(\varphi_K(x_i) - \mu_K^-)^T.$$

Note that (2) is a supervised learning model which neglects the knowledge of the unlabeled data, and hence can not yield the favorable classification. In addition, this model requires all samples to come from the identical distribution, which means that it can not deal with the cross-domain problem. Motivated by this, we propose a domain transfer kernel learning method in the next subsection.

3.2 Domain Transfer Kernel Learning for Discriminant Analysis

Note that if the distributions $\mathcal{P}(x)$ and $\mathcal{Q}(x)$ are completely independent, then the out-of-domain data $\mathcal{D}^{out}$ is useless; if $\mathcal{P}(x)$ and $\mathcal{Q}(x)$ are identical, then the cross-domain problem becomes the standard classification problem. However, in most cases $\mathcal{P}(x)$ and $\mathcal{Q}(x)$ are neither independent nor identical, for which we may use the cross-domain projection vector that is learned from out-of-domain data set $\mathcal{D}^{out}$ and the in-domain data set $\mathcal{D}^{in}$ with MMD formulation. Then, the optimization problem of DTDKL can be formulated as:

$$\max_{w,K} \quad \mathrm{KLDA}_{K,w}(\mathcal{D}_l) - \beta \mathrm{MMD}_K^2(\mathcal{D}^{out}, \mathcal{D}^{in}), \tag{3}$$

where $\beta \geq 0$ is a parameter to balance the difference of data distributions of two domains and the Fisher discriminant ratio of KLDA for labeled samples. This optimization problem involves two classes of variables. One is the kernel matrix K which represents the adaptive feature space, and the other is the projection direction w for the dimensionality reduction.

By specializing $\mathrm{KLDA}_{K,w}(\mathcal{D}_l)$ and $\mathrm{MMD}_K^2(\mathcal{D}^{out}, \mathcal{D}^{in})$ as $F(w, K)$ and $\|\mu^{in} - \mu^{out}\|$, respectively, (3) becomes

$$\begin{aligned} &\max_{w,K} \quad F_{\lambda,\beta}(w, K) \\ &\text{s.t.} \quad K = \sum_{i=1}^{p}\theta_i K_i,\ \mathbf{1}^T\theta = 1,\ \theta \succeq 0 \end{aligned} \tag{4}$$

where

$$F_{\lambda,\beta}(w, K) = \frac{(w^T(\mu_K^+ - \mu_K^-))^2}{w^T(\Sigma_K^+ + \Sigma_K^- + \lambda I)w} - \beta \left\|\mu_K^{in} - \mu_K^{out}\right\|^2. \tag{5}$$

and $\mu_K^+, \mu_K^-, \Sigma_K^+, \Sigma_K^-$ denote the training samples' (in-domain and out-of-domain) means and covariances, respectively. Comparing with the model (2), we see that a new term $-\|\mu^{in} - \mu^{out}\|^2$ is introduced into the objective of (4). This term is

a concave function that will bring a concavification effect on the original non-concave objective of (2). So, the globally optimal solution of the maximization problem (4) can be easier found than that of the problem (2) proposed in [9] .

Note that the last term in $F_{\lambda,\beta}(w,K)$ is independent of w. Hence, the maximization problem (4) can be rewritten as

$$\begin{aligned} &\max_{K} \ \max_{w} \ \left\{ \frac{(w^T(\mu_K^+ - \mu_K^-))^2}{w^T(\Sigma_K^+ + \Sigma_K^- + \lambda I)w} \right\} - \beta \left\| \mu_K^{in} - \mu_K^{out} \right\|^2 \\ &\text{s.t.} \quad K = \sum_{i=1}^{p} \theta_i K_i, \ \mathbf{1}^T\theta = 1, \ \theta \succeq 0. \end{aligned} \tag{6}$$

Using the same arguments as in [9], we know that the globally optimal solution of the inner maximization problem

$$w^* = (\Sigma_K^+ + \Sigma_K^- + \lambda I)^{-1}(\mu_K^+ - \mu_K^-). \tag{7}$$

Substituting this into the objective of (8), we obtain that

$$\begin{aligned} &\max_{K} \ F_{\lambda,\beta}^*(K) \\ &\text{s.t.} \quad K = \sum_{i=1}^{p} \theta_i K_i, \ \mathbf{1}^T\theta = 1, \ \theta \succeq 0 \end{aligned} \tag{8}$$

where

$$\begin{aligned} F_{\lambda,\beta}^*(K) = &(\mu_K^+ - \mu_K^-)^T(\Sigma_K^+ + \Sigma_K^- + \lambda I)^{-1}(\mu_K^+ - \mu_K^-) \\ &-\beta \left\| \mu_K^{in} - \mu_K^{out} \right\|^2 . \end{aligned} \tag{9}$$

On the other hand, from the Representer Theory [12], the optimal discriminative projection in DTDKL is the span of the images of the training points in the feature space. Note that in this method the training set includes both labeled data and unlabeled data due to the MMD formulation. Hence, there exists a vector $\alpha \in \mathbb{R}^n$ such that

$$w^* = \sum_{i=1}^{n} \alpha_i^* \varphi_K(x_i) = U_K \alpha^* \tag{10}$$

where

$$U_K = [\varphi_K(x_1) \ \cdots \ \varphi_K(x_n)] .$$

In fact, we can find a closed-form expression of α:

$$\alpha^* = \frac{1}{\lambda}[I - G(\lambda I + GKG)^{-1}GK]a \tag{11}$$

where a is an n-dimensional vector given by

$$a = [1/n_+, \ldots, 1/n_+, -1/n_-, \ldots, -1/n_-, 0, \ldots 0]^T \in \mathbb{R}^n,$$

and the matrix G is defined as

$$G = \begin{pmatrix} \frac{1}{\sqrt{n_+}}(I - \frac{1}{n_+}\mathbf{1}_{n_+}\mathbf{1}_{n_+}^T) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{n_-}}(I - \frac{1}{n_-}\mathbf{1}_{n_-}\mathbf{1}_{n_-}^T) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

Because the unlabel data is introduced in this model, the representation of the variables are different from those of [9]. By equations (7), (10) and (11), $F^*_{\lambda,\beta}(K)$ can be written as

$$\begin{aligned} F^*_{\lambda,\beta}(K) &= (\mu_K^+ - \mu_K^-)^T(\Sigma_K^+ + \Sigma_K^- + \lambda I)^{-1}(\mu_K^+ - \mu_K^-) \\ &\quad -\beta b^T K b \\ &= (\mu_K^+ - \mu_K^-)^T w^* - \beta b^T K b \\ &= a^T U_K^T U_K \alpha^* - \beta b^T K b \\ &= \frac{1}{\lambda} a^T K (I - G(\lambda I + GKG)^{-1} GK) a - \beta b^T K b \end{aligned}$$

where $b = (b_1, \ldots, b_n)$ with

$$b_i = \begin{cases} \frac{1}{n^{out}} & \text{if} \quad x_i \in \mathcal{D}^{out}; \\ -\frac{1}{n^{in}} & \text{if} \quad x_i \in \mathcal{D}^{in}. \end{cases} \tag{12}$$

Then, the optimization problem (8) can be reformulated as

$$\begin{aligned} \min_{\theta,t} \ & -\frac{1}{\lambda}\sum_{i=1}^{p} \theta_i (a^T K_i a - \lambda\beta b^T K_i b) + t \\ \text{s.t.} \quad & a^T KG(\lambda I + GKG)^{-1} GKa \le t, \\ & \mathbf{1}^T\theta = 1, \ \theta \succeq 0. \end{aligned} \tag{13}$$

By the Schur Complement Theorem, we know that

$$a^T KG(\lambda I + GKG)^{-1} GKa \le t \Leftrightarrow \begin{pmatrix} \lambda I + GKG & GKa \\ a^T KG & t \end{pmatrix} \succeq 0.$$

The last two equations show that (13) is equivalent to

$$\begin{aligned} \min_{t\in\mathbb{R},\theta\in\mathbb{R}^p} \ & \frac{1}{\lambda}\left(t - \sum_{i=1}^{p} \theta_i (a + \sqrt{\beta}b)^T K_i (a + \sqrt{\beta}b)\right) \\ \text{s.t.} \quad & S(t,\theta) \succeq 0 \\ & \mathbf{1}^T\theta = 1, \quad \theta \succeq 0, \end{aligned} \tag{14}$$

where

$$S(t,\theta) = \begin{pmatrix} \sum_{i=1}^{p} \theta_i J^T K_i G + \lambda I & \sum_{i=1}^{p} \theta_i G^T K_i a \\ \sum_{i=1}^{p} \theta_i a^T K_i G & t \end{pmatrix}.$$

Algorithm 1. Kernel Discriminant Learning in Cross-domain Problem

input : A labeled out-of-domain data set $\mathcal{D}^{out} = \{x_i^{out}, y_i^{out}\}$, an unlabeled in-domain data set $\mathcal{D}^{in} = \{x_i^{in}\}$ and positive parameters λ, β.

output: Labels Y^{in} of the unlabeled data X^{in} in the in-domain.

1. Solve SDP problem in (14) to obtain a kernel matrix K
2. Compute the coefficient vector α through (11), then the direction w can be obtained from (10)
3. Use the direction w to get new representations $\{x_i^{out'}\}$ and $\{x_i^{in'}\}$ of the original data $\{x_i^{out}\}$ and $\{x_i^{in}\}$, respectively.
4. Train a classifier or regressor: $f : x_i^{out'} \rightarrow y_i^{out'}$
5. Use the trained classifier or regressor to predict the labels

Thus, we convert the nonconvex optimization problem (4) into a convex semidefinite programming problem. Similar to the one obtained by [9], it can be solved by interior-point method softwares such as SeDuMi or SDPT3.

The cost of constructing the basic kernel matrices is $O(n^2 d)$, the combining the p basic kernel matrices costs $O(n^2 p)$, and computing the gradient and Hessian of the objective is $O(n^3)$. so the total cost per Newton step of interior-point methods which can solve SDP is $O(p^3 + n^2 d + n^2 p + n^3)$. In the case of $p, d \ll n$, the total cost grows like $O(n^3)$, which is the same as that of SVMs. The SDP guarantees the convergence of the algorithm.

4 Experiment

In this work, we carried out experiments on three real-world data collections from two different domains to evaluate the described algorithms. The performance is compared with MKDL-DA [9], and Semi-supervised kernel discriminant analysis SKDA [13] as well as other transferred dimensionality reduction method, TKDR [2] and MMDE [3].

4.1 Data Sets and Experiment Setup

As shown in Table 1, the data collections consist of Reuters-21578 [14], 20-Newsgroups [15] and SyskillWebert [14]. Amoung them, Reuters-21578 and 20 Newsgroups is the standard used to test web page ratings. The important statistics and pre-processing procedures of these collections are presented below.

Data Sets Description. With a hierarchical structure, SyskillWebert database consists of the HTML source of web pages plus the ratings of a user on those web pages. Four separate subjects are contained in the web pages. Associated with each web page are the HTML source and a user's rating in terms of "hot", "medium" or "cold" [16]. As demonstrated in Table 1, all of the four subjects are involved in our study. "Goat" is reserved as the set of in-domain and the other are used as the out-of-domain data. Compared to the "cold" pages, the

total number of pages rated as "medium" or "hot" is fewer. Hence, we combine the "medium" and "hot" pages together, and change the labels of those pages as "non-cold" to form a binary classification problem. The learning task is to predict the user's preferences for the given web pages. the Rueters-21578 is another text repository which consists of Reuters news wire articles organized into five top categories, and each category contains various sub-categories. Three categories, "orgs", "people" and "places", we remove all the documents of "USA" in order to make the size of these three categories nearly even [16]. For each category, all of the sub-categories are then organized into two parts, and each part has different distribution and approximately equal size. Therefore, one part can be used for the in-domain and the other is treated as the out-of-domain purpose. According to the method described in [17], three cross-domain learning tasks are generated as listed in Table 1, and the learning objective aims to classify articles into top categories. Similar to Reuters-21578 data, 20-Newsgroups corpus contains 7 top categories and these top categories contain 20 subcategories which have approximately 20,000 newsgroup documents. We select four top categories "com", "rec", "talk" and "sci" in this experiment. Thus, three other cross-domain tasks are formed as listed in Table 1.

Table 1. Summary of Datasets

Data Set		In-domain	Out-of-domain
SyskillWebert		Goat	Bands Sheep Biomedical
Reuters	Orgs vs People Orgs vs Places People vs Places	Documents of some sub categories	Documents of other sub categories
20 Newsgroup	Com vs Rec Rec vs Sci Rec vs Talk	Documents of some sub categories	Documents of other sub categories

Experiment Setup. On one hand, for each in-domain data set employed in the experiment, we further split it into two parts: in-domain data with labels($\mathcal{D}_l$) and the in-domain data without labels($\mathcal{D}_u$). We randomly select 50% data from out-of-domain and in-domain, respectively. The ratio between $|\mathcal{D}_l|$ and $|\mathcal{D}_u|$ is 1:9. All of the in-domain data without labels($\mathcal{D}_u$) are used as the test sets while the training sets consist of the data points with labels from both the in-domain $\mathcal{D}_l$ and out-of-domain ($\mathcal{D}^{out}$). On the other hand, the kernel is a convex combination of 10 Gaussian kernels [10]:

$$K(x,z) = \sum_{i=1}^{10} \theta_i e^{\|x-z\|^2/\sigma_i^2}$$

where θ_i are the weights of the kernels to be determined. The values of σ_i were chosen uniformly over the interval $[10^{-1}, 10^2]$ on the logarithmic scale. The

regularization parameter λ in DTDKL and the MMD parameter β was fixed to 10^{-6} and 1, respectively. As a matter of fact, the algorithm is not sensitive to the parameter β for a wide range.

Any ordinary classifier, such as Naïve Bayes, K-nearest, can be used in the dimensionality reduction method. In our experiments, we simply choose the nearest centroid method.

4.2 Experimental Results

For performance evaluation, we use accuracy, which has been widely used as a evaluation metric, we systematically compare the proposed algorithms to some classifiers, including discriminant MKL-DA [9], SKDA [13], as well as TKDR [2], MMDE[3]. All of the results reported below are mean of that running 10 times.

Table 2. Comparison of Performance (*mean* ± *std %)*

Data Set		MKL-DA	SKDA	TKDR	MMDE	DTDKL
Syskill - Webert	Goat-Bands	50.47 (11.31)	55.56 (1.63)	54.81 (1.24)	57.98 (4.52)	**59.03 (1.13)**
	Goat-Biomedical	58.38 (8.40)	60.54 (8.60)	61.14 (1.13)	60.32 (2.21)	**61.38 (1.30)**
	Goat-Sheep	80.52 (2.25)	67.38 (9.58)	71.05 (3.62)	80.04 (1.33)	**81.75 (1.65)**
Reuters	Orgs-People	67.08 (5.96)	66.05 (6.02)	68.80 (4.81)	71.22 (3.14)	**73.65 (5.05)**
	Orgs-Places	54.90 (8.56)	55.60 (4.91)	58.31 (4.98)	69.19 (3.31)	**70.90 (1.98)**
	People-Places	50.54 (8.98)	54.94 (4.32)	53.60 (4.80)	64.86 (4.01)	**65.60 (4.98)**
20 Newsgroup	Com-Rec	54.49 (9.05)	72.02 (7.42)	72.69 (7.29)	73.05 (3.98)	**73.46 (3.32)**
	Rec-Sci	70.05 (8.78)	73.90 (4.06)	76.90 (5.53)	77.63 (3.29)	**77.68 (4.85)**
	Rec-Talk	70.80 (2.96)	77.35 (5.77)	78.03 (6.64)	78.90 (2.18)	**79.08 (1.52)**

In this section, we use accuracy as the evaluation metric, and compare the proposed algorithms to MKL-DA, SKDA and TKDR. The results show clearly that DTDKL is able to alleviate the influence of different distributions.

Table 2 summarizes the accuracies of MKL-DA, SKDA, TKDR, MMDE, and DTDKL on the three databases with the best results highlighted in bold font. It can be seen that the MAP of the DTDKL methods is consistently higher than the other methods on all of the data sets. Moreover, it is general trend that those problems with higher precision, generally, have a smaller error.

Overall Performance

Our proposed method DTDKL outperforms all the other algorithms in terms of accuracy, demonstrating that DTDKL learns a robust target classifier.And it is also easy to notice that the MMDE and DTDKL performs much better than the other three methods, even TKDR. Moreover, the standard deviation of DTDKL is much smaller, means that it is more stable. For the SyskillWebert collection, compared to DTDKL's rivals, on average it achieves at least 1.23%, 0.24% and 1.05% higher accuracy on "GoatVsBands", "GoatsVsBiomedical" and "GoatVsSheep", respectively. The better performance can be ascribed to transferring the in-domain and out-domain data to a features whose the discriminant

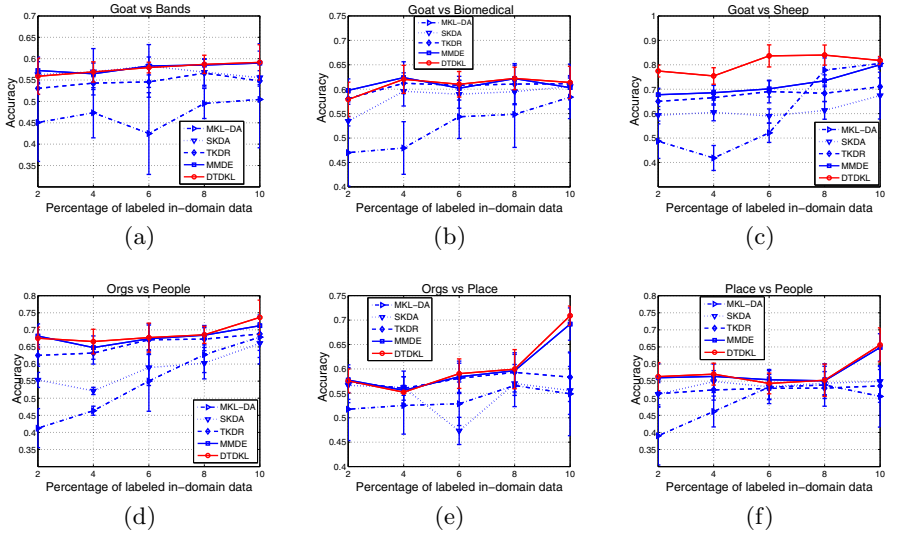


Fig. 1. Accuracy vs. different size of $\mathcal{D}_l^{in}$

distance of the data is maximum and the maximum mean discrepancy comes out to be minimum. For the Reuters-21578 data set, the accuracy of DTDKL on average achieve at least 1.6%, 1.7% and 0.7% higher that other approaches on "OrgsVsPeople", "OrgsVsPlaces" and "PeopleVsPlaces" respectively.The similar performance explanation provided to DTDKL method on Reuters-21578 can also applied here. On the 20 News-group data set, the DTDKL methods perform best among the total tasks. Compare DTDKL and MKL-DA, we can see that the MAP of DTDKL is at least 6.6% higher, even nearly 10% higher than MKL-DA, which confirm the positive effect of MMD.

Sensitivity

This study evaluates the sensitivity of varied sizes of labeled in-domain data and conducted on the three collection. The results are demonstrated in Fig.1. It is evident that, as the size of the labeled in-domain data increases, DTDKL performs better than or as equal as its competitors at most case. For example, as shown in Figure 1(c), DTDKL achieves at least 5% higher accuracy than other methods on each size of labeled in-domain data. As a general trend, the accuracy of DTDKL steadily improves when the number of labeled in-domain data increase from 1% to 10%. Consequently, we infer that, better performances can be obtained if more labeled in-domain data are provided.

5 Conclusion

We have proposed a unified dimensionality reduction in cross-domain problems to simultaneously learn a kernel function as well as Fisher discriminant direction by maximizing the Fisher discriminant distance and minimizing the distance of

out-of-domain and in-domain. Moreover, we assume that the kernel function in optimal kernel discriminant analysis is a linear combination of multiple base kernels; Thus, it can be efficiently solve by SDP. Experimental result show that DTDKL method outperforms existing dimensionality reduction in cross-domain in three text data sets.

References

1. Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mullers, K.: Fisher discriminant analysis with kernels. In: NNSP Workshop, pp. 41–48 (1999)
2. Wang, Z., Song, Y., Zhang, C.: Transferred Dimensionality Reduction. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 550–565. Springer, Heidelberg (2008)
3. Pan, S., Kwok, J., Yang, Q.: Transfer learning via dimensionality reduction. In: AI, vol. 2, pp. 677–682 (2008)
4. Si, S., Tao, D., Chan, K.: Evolutionary cross-domain discriminative hessian eigenmaps. IEEE Transactions on Image Processing 19(4), 1075–1086 (2010)
5. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.: Face recognition using laplacianfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(3), 328–340 (2005)
6. Fukunaga, K.: Introduction to statistical pattern recognition. Academic Pr. (1990)
7. Baudat, G., Anouar, F.: Generalized discriminant analysis using a kernel approach. Neural Computation 12(10), 2385–2404 (2000)
8. Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L., Jordan, M.: Learning the kernel matrix with semidefinite programming. The Journal of Machine Learning Research 5, 27–72 (2004)
9. Kim, S.J., Magnani, A., Boyd, S.: Optimal kernel selection in kernel fisher discriminant analysis. In: ICML, pp. 465–472 (2006)
10. Ye, J., Ji, S., Chen, J.: Multi-class discriminant kernel learning via convex programming. The Journal of Machine Learning Research 9, 719–758 (2008)
11. Borgwardt, K., Gretton, A., Rasch, M., Kriegel, H., Schölkopf, B., Smola, A.: Integrating structured biological data by kernel maximum mean discrepancy. Bioinformatics 22(14), e49–e57 (2006)
12. Cristianini, N., Shawe-Taylor, J.: Kernel methods for pattern analysis. Cambridge University Press, Cambridge (2004)
13. Cai, D., He, X., Han, J.: Semi-supervised discriminant analysis. In: ICCV, pp. 1–7 (2007)
14. Asuncioin, A., Newman, D.: Uci machine learning repository (2007), http://www.ics.uci.edu/mlearn/MLRepository.html
15. Davidov, D., Gabrilovich, E., Markovitch, S.: Parameterized generation of labeled datasets for text categorization based on a hierarchical directory. In: SIGIR, pp. 250–257 (2004)
16. Zhong, E., Fan, W., Peng, J., Zhang, K., Ren, J., Turaga, D., Verscheure, O.: Cross domain distribution adaptation via kernel mapping. In: SIGKDD, pp. 1027–1036 (2009)
17. Dai, W., Yang, Q., Xue, G., Yu, Y.: Boosting for transfer learning. In: ICML, pp. 193–200 (2007)

Prioritizing Disease Genes by Bi-Random Walk

Maoqiang Xie[1], Taehyun Hwang[2], and Rui Kuang[3,⋆]

[1] College of Software, Nankai University, Tianjin, China
[2] Masonic Cancer Center, University of Minnesota, Twin Cities, USA
[3] Department of Computer Science and Engineering,
University of Minnesota, Twin Cities, USA
kuang@cs.umn.edu

Abstract. Random walk methods have been successfully applied to prioritizing disease causal genes. In this paper, we propose a bi-random walk algorithm (BiRW) based on a regularization framework for graph matching to globally prioritize disease genes for all phenotypes simultaneously. While previous methods perform random walk either on the protein-protein interaction network or the complete phenome-genome heterogenous network, BiRW performs random walk on the Kronecker product graph between the protein-protein interaction network and the phenotype similarity network. Three variations of BiRW that perform balanced or unbalanced bi-directional random walks are analyzed and compared with other random walk methods. Experiments on analyzing the disease phenotype-gene associations in Online Mendelian Inheritance in Man (OMIM) demonstrate that BiRW effectively improved disease gene prioritization over existing methods by ranking more known associations in the top 100 out of nearly 10,000 candidate genes.

Keywords: Disease Gene Prioritization, Bi-Random Walk, Graph-based Learning.

1 Introduction

It is now well accepted that phenotypes are determined by genetic material under environmental influences. To understand the relation between disease phenotypes and genes, numerous genomic studies on large patient cohorts such as genome-wide association studies [1] [2] have been conducted to identify candidate disease genes, and in the past decade, the knowledge of determined disease phenotype-gene associations has been quickly accumulated in databases such as Online Mendelian Inheritance in Man (OMIM), a database of human genes and genetic disorders. Driven by the accumulated knowledge, random walk-based algorithms, which take the advantage of the availability of large phenotypic and molecular networks (Fig. 1), were proposed to utilize the disease modules and gene modules in the networks to prioritize disease genes [3] [4] [5] [6] [7] [8] [9]. The human disease phenotype network [10] provides information on phenotype similarities computed

⋆ Corresponding author.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 292–303, 2012.

by text mining of the full text and clinical synopsis of the disease phenotypes in OMIM [11]. Large molecular networks such as the human protein-protein interaction network [12] [13] or functional linkage network [6] provide functional relations among genes. Based on the observation that genes associated with the same or related diseases tend to interact with each other in the gene network and similar phenotypes tend to share the same disease genes, random walk provides an effective framework to explore the relations in the networks.

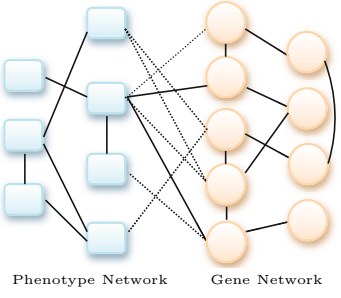


Fig. 1. Predicting missing associations in disease phenotype-gene association network. The solid and dash lines represent known and missing associations, respectively.

Motived by the graph matching problem, we postulate that phenotype-gene associations can be characterized by paired associations between close by genes in the PPI network and close by phenotypes in the phenotype similarity network. Confirmed by the high frequency of such paired associations in OMIM, we propose a bi-random walk algorithm (BiRW) to capture the patterns in the networks to unveil the association between the complete collection of disease phenotypes and genes (phenome-genome association). The key assumption is that the global structure of phenome-genome association can be represented by paired associations, and thus, the reconstruction of the complete phenome-genome association can be achieved by maximizing the number of such paired associations constrained on the known associations. BiRW algorithm iteratively adds new associations into the network by bi-random walk to evaluate the number of recovered paired associations with a decay factor penalizing the number of steps. We investigated variants of BiRW by performing bi-random walk with balanced or unbalanced steps in the the PPI network and the phenotype similarity network, and evaluated the methods by experiments on OMIM data.

2 Methods

The disease phenotype-gene association network (or phenome-genome association network) is a heterogeneous network composed of a phenotype network, a gene network and the known phenotype-gene associations modeled by a bipartite graph (Fig. 1). Let $P_{(m\times m)}$, $G_{(n\times n)}$ and $A_{(m\times n)}$ be the adjacency matrix of the phenotype network, the gene network and the association bipartite graph respectively, where m is the number of phenotypes and n is the number of genes. The objective is to predict the missing associations based on the heterogenous disease phenotype-gene association network by reconstructing an association matrix $R_{(m\times n)}$. The magnitude of each R_{ij} provides the degree of association between phenotype i and gene j. In the following, we first introduce the loss function for the learning problem and then the Bi-Random Walk algorithm (BiRW) that minimizes the cost function for learning R.

2.1 Loss Function

Our assumption is that similar (or the same) phenotypes are more likely to share the same causal gene or causal genes that interact with each other. More specifically, we assume that the predicted paired associations should form the following subgraph patterns: 1) the triangle with two phenotype nodes and one gene node following the assumption "similar phenotypes may share the same causal gene", 2) the triangle with one phenotype node and two gene nodes following the assumption "causal genes of the same disease phenotype tend to interact", and 3) the rectangle with two phenotype nodes and two gene nodes following the assumption "genes associated with similar phenotypes tend to interact". Based on the assumptions, we define the following loss function over R,

$$L(R) = \alpha \sum_{u,v,i,j} (P \otimes G)_{(i,u),(j,v)} (R_{i,u} - R_{j,v})^2 + (1-\alpha) \sum_{i,u} (R_{i,u} - A_{i,u})^2,$$

where $P \otimes G$ is the Kronecker product of P and G. Each $P \otimes G_{(i,u),(j,v)}$ is 1 if $P_{i,j} = 1$ and $G_{u,v} = 1$, in other words phenotype i and j are neighbors and gene u and v are also neighbors, and otherwise 0. In this loss function, the first term enforces a smoothness on R where phenotypes (i, j) and gene (u, v) should form paired associations with phenotype i aligned with gene u and phenotype j aligned with gene v when (i, j) are neighbors and (u, v) are also neighbors. The second term uses prior knowledge A as a regularization term. The trade-off between these two competing constraints is controlled by a positive parameter $\alpha \in (0, 1]$. Intuitively, the cost function in equation (1) evaluates that by associating a phenotype and a gene in R, how many paired associations are curated. The interpretation is closely related to global network alignment algorithms that were applied to align protein-protein interaction networks across species [14] [15] [16] [17] [18]. Since the first term is actually a quadratic term of the elements in R with Hessian $D - (P \otimes G)$, the Laplacian of graph $P \otimes G$, the loss function can be rewritten as the following quadratic function,

$$\min_{R} \quad \alpha \overrightarrow{R}^T (D - (P \otimes G)) \overrightarrow{R} + (1-\alpha) || \overrightarrow{R} - \overrightarrow{A} ||^2, \tag{1}$$

where $\overrightarrow{R}$ is the vector concatenated from the rows in R and D is the diagonal matrix with the row sum of $P \otimes G$ as the diagonal entries.

2.2 Bi-Random Walk

To minimize the loss function in equation (1), a straightforward method is to apply random walk with restart on the Kronecker product matrix $P \otimes G$. Since $P \otimes G$ is $(m \times n)$ by $(m \times n)$, this approach does not scale to the large network. We propose a bi-random walk strategy (BiRW), which performs random walk on the phenotype network and the gene network simultaneously. BiRW aims to maximize the number of paired associations by bi-random walk on both phenotype network and gene network to evaluate potential candidate associations

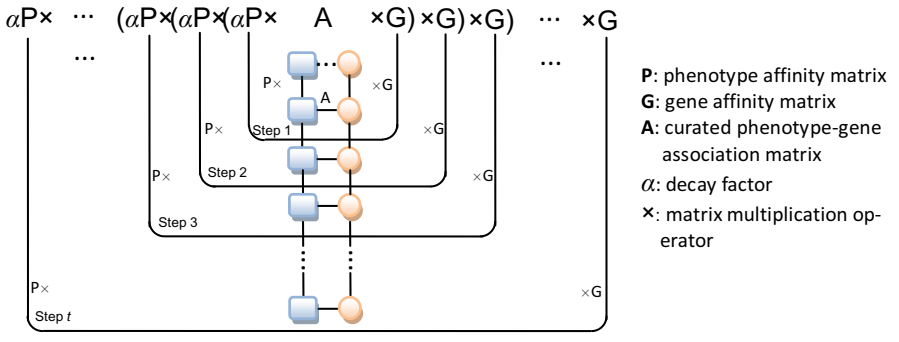


Fig. 2. Illustration of bi-random walk algorithm. P and G are the affinity matrices of the phenotype network and the gene network, respectively. A is the bipartite graph of the known phenotype-gene association from OMIM. By iteratively extending the phenotype path and the gene path (achieved by multiplying P on the left or G on the right in each step), the algorithm maximizes the number of paired associations (loops between phenotypes and genes) with the steps weighted by a decay factor $\alpha \in (0, 1)$. The dashed edge indicates a potential association to add into the network. The iterative algorithm finds the number of new paired associations formed by introducing this additional connection.

(Fig. 2). By iteratively extending the phenotype path and the gene path (achieved by multiplying P on the left and G on the right in each step), the algorithm evaluates each candidate association by the number of closed loops weighted by a decay factor $\alpha \in (0, 1)$. The decay factor down-weights the importance of newly formed loops as the number of random walk steps is getting larger. Here, the matrix multiplications $(P_{(m\times m)} \cdot A_{(m\times n)} \cdot G_{(n\times n)})$ mimic jumps on the phenotype network, the gene network and the association network. In the first step, each element $(P \cdot A \cdot G)_{(i,j)}$ represents the number of paired associations obtained by connecting a target phenotype i to a candidate gene j with phenotype or gene paths length 1. If we ignore the decay factor for now, more generally, after t steps of multiplication $P \ldots (P \cdot (P \cdot A \cdot G) \cdot G) \ldots G = P^t \cdot A \cdot G^t$, the loop patterns curated with up to t steps of random walks can be evaluated. To achieve the best solution $R_{(m\times n)}$, we formulated the problem as $R = P \cdot R \cdot G$, assuming P is column-normalized, G is row-normalized, and the elements in R add to 1. $P \cdot R \cdot G$ can be rewritten in a vector form $P \otimes G\overrightarrow{R}$. Each bi-random walk is the same as a random walk on the Markov matrix $P \otimes G$. Thus, applying bi-random walk is identical to using power method to find the stationary distribution of $P \otimes G$. Note that the idea is also similar to a normalized and relaxed version of regular graph-matching methods [17], which maximize the number of matched edges in two graphs (the phenotype network and the gene network). In addition, the known OMIM associations A normalized the same as R is introduced as priori knowledge. The complete form of the model is as follows,

$$R = \alpha P \cdot R \cdot G + (1 - \alpha)A, \tag{2}$$

The decay factor α also plays the role to balance the objective of closed loops for evaluating candidate associations and the consistence with the known associations in A. This equation can be solved by iteratively updating R by calculating the right side of the equation (2) with the current R. The process also converges to a unique solution [18]. Candidate associations can then be selected by the magnitude of the scores in R. Essentially, this algorithm is mathematically equivalent to the label propagation algorithm in [19], and it was shown that the algorithm minimizes the cost function in equation (1).

2.3 Unbalanced Bi-Random Walk

As illustrated in Fig. 2, the steps to walk on the phenotype network and the gene network explicitly summarize the closed loops in the previous step. Theoretically, the random walk in the two directions will eventually converge to a stationary distribution as the unique solution. However, since only the closed loops of smaller path lengths are informative for predicting associations, excessively counting loops obtained by a large number of random walk steps could introduce false positives. Moreover, the phenotype similarity network and the gene network contain different topologies and structures, and thus, the optimal number of random walk steps might be different on the two networks. To address the problem, we restrict the number of random walk steps on the two sides by introducing two additional parameters l and r as the numbers of maximal iterations in the following left/right random walk on the networks,

$$\begin{aligned} \text{Left Walk:} \quad & R_t = \alpha P \cdot R_{t-1} + (1-\alpha)A \\ \text{Right Walk:} \quad & R_t = \alpha R_{t-1} \cdot G + (1-\alpha)A \end{aligned} \tag{3}$$

Left Walk and Right Walk could be applied alternatively to introduce additional steps in either phenotype network or gene network. The new formula does not converge as equation 2 to a closed-form but it carries the same interpretation that each left or right walk extends either the phenotype path length or the gene path length. Empirically, l, r and α are the parameters tuned by cross-validation on the training data.

2.4 BiRW Algorithms

Given phenotype network P, gene network G, and the phenotype-gene associations A, we first normalize the matrices $\bar{P} = D_P^{-\frac{1}{2}} \cdot P \cdot D_P^{-\frac{1}{2}}$ and $\bar{G} = D_G^{-\frac{1}{2}} \cdot G \cdot D_G^{-\frac{1}{2}}$, where D_P is a diagonal matrix with diagonal elements $D_{Pii} = \sum_j P_{ij}$, and $\bar{G}$ is the same normalized from G. Depending on the arrangement of the left/right walk, we consider three variations of BiRW.

BiRW_bl: This algorithm exactly implements the balanced BiRW given in equation (2), and computes the closed-form solution of equation (1).

BiRW_bl($\bar{P}, \bar{G}, A, \alpha$)
1 $R_0 = \frac{A}{sum(A)}, t = 1$
2 **Do until convege**
3 $R_t = \alpha \bar{P} \cdot R_{t-1} \cdot \bar{G} + (1-\alpha)A$
4 $t = t + 1$
5 **return** (R)

BiRW_avg: This algorithm implements the unbalanced BiRW with the averaged output from the left walk and the right walk in each step.

BiRW_avg($\bar{P}, \bar{G}, A, \alpha, l, r$)
1 $R_0 = \frac{A}{sum(A)}$
2 **for** $t = 1$ **to** $max(l, r)$
3 **if** $t <= l$
4 $R_{t\_left} = \alpha \bar{P} \cdot R_{t-1} + (1-\alpha)A$
5 **if** $t <= r$
6 $R_{t\_right} = \alpha R_{t-1} \cdot \bar{G} + (1-\alpha)A$
7 $R_t = (\delta_{t \le r} \cdot R_{t\_left} + \delta_{t \le l} \cdot R_{t\_right})/(\delta_{t \le l} + \delta_{t \le r})$
8 **return** (R)

In the algorithm, $\delta_{t \le x}$ is 1 if $t \le x$ and 0 otherwise.

BiRW_seq: This algorithm implements the unbalanced BiRW with sequential walk with left walk followed by right walk in each step.

BiRW_seq($\bar{P}, \bar{G}, A, \alpha, l, r$)
1 $R_0 = A = \frac{A}{sum(A)}$
2 **for** $t = 1$ **to** $max(l, r)$
3 **if** $t <= l$
4 $R_{t\_left} = \alpha \bar{P} \cdot R_{t-1} + (1-\alpha)A$
5 **if** $t <= r$
6 $R_t = \alpha R_{t\_left} \cdot \bar{G} + (1-\alpha)A$
7 **return** (R)

3 Comparison of Random Walk Algorithms

In this section, we compare BiRW with the other random walk or label propagation algorithms for disease gene prioritization [4] [6] [7] [8] [9]. For example, PRINCE performs label propagation on the PPI network to prioritize disease genes [8]. The initial probabilities on the gene nodes are normalized from the causative genes of the nearest neighbors of the query phenotype p chosen by a logistic function. The initial scores are propagated in the stochastic matrix normalized from the PPI network. After convergence, the unique solution of label propagation is used to rank the genes. RWRH [9] runs the same label propagation algorithm on the combined heterogeneous network of all the three networks to rank genes for a query phenotype. MINProp [7] is based on a principled way to integrate three networks in an optimization framework and performs iterative label propagation on each individual subnetwork. These disease gene prioritization

algorithms rank genes based on their predicted association against a particular query phenotype while BiRW is a global approach which identifies the missing associations of all the phenotypes simultaneously. Thus, conceptually, BiRW is a phenome-genome approach while the other algorithms are phenotype-wise approaches, none of which explores the relation between the predicted associations across the phenotypes. To illustrate the difference between BiRW and the other methods, we compared the initialization and the random walk steps of the algorithms in Table 1. The first difference is that these methods learn with the structure of different networks. Random Walk, Diffusion Kernel and PRINCE perform random walk only on the PPI network combined with the direct neighbors inferred from the phenotype network and the known associations. RWRH and MINProp perform random walk on the complete heterogenous phenome-genome association network. BiRW performs random walk on the Kronecker product graph of the phenotype network and the gene network in the balanced case or on the phenotype network and the gene network separately in the unbalanced case.

Table 1. Comparison of random walk algorithms for disease gene prioritization. We denote the target variables for assigning prediction scores on the phenotype nodes and the gene nodes $p_{(m\times 1)}$ and $g_{(n\times 1)}$, respectively. q is the index of the query phenotype. For any matrix X, $\hat{X}$ represents the row normalized stochastic matrix from X. α, β and λ are positive parameters $\in (0, 1)$.

Algorithm	Initialization and Random walk step(s)
Random Walk [4] [6]	$g^0 = (A_{q*})'$ $g^t = \alpha G g^{t-1} + (1-\alpha) g^0$
Diffusion Kernel [4]	$g^0 = (A_{q*})'$ $g = (e^{-\beta(D_G - G)}) * g^0$
PRINCE [8]	$g^0(i) = logit(max_l(P_{ql} * A_{li}))$ $g^t = \alpha \bar{G} g^{t-1} + (1-\alpha) g^0$
RWRH [9]	$g^0 = 0, \begin{cases} p^0(i) = 0, \forall i \neq q \\ p^0(q) = 1 \end{cases}$ $\begin{pmatrix} p^t \\ g^t \end{pmatrix} = \alpha \begin{pmatrix} (1-\lambda)\hat{P} & \lambda\hat{A} \\ \lambda\hat{A}^T & (1-\lambda)\hat{G} \end{pmatrix} \begin{pmatrix} p^{t-1} \\ g^{t-1} \end{pmatrix} + (1-\alpha) \begin{pmatrix} p^0 \\ g^0 \end{pmatrix}$
MINProp [7]	$g^0 = 0, \begin{cases} p^0(i) = 0, \ \forall i \neq q \\ p^0(q) = 1, \end{cases}$ Repeat to solve two random walk problems until converge 1) $p^t = \beta \bar{P} p^{t-1} + (1-\beta)(\frac{1-2\beta}{1-\beta} p^0 + \frac{\beta}{1-\beta} \bar{A} g)$ 2) $g^t = \alpha \bar{G} g^{t-1} + (1-\alpha)(\frac{1-2\alpha}{1-\alpha} g^0 + \frac{\alpha}{1-\alpha} \bar{A}' p)$
BiRW	$R = 0$ $R^t = \alpha \bar{P} R^{t-1} \bar{G} + (1-\alpha) * \bar{A}$

Another mathematical difference between BiRW and the other algorithms lies in the formulation of using the known associations in A. PRINCE uses the known associations to decide an initial set of genes that are associated with a query phenotype. RWRH and MINProp directly use A as part of the large network for

random walk. BiRW treats R as the target variable and the known association A as a regularization of R, intuitively, because A is only partially known and most of the zero entries of A are "unknown" instead of "no association". Thus, using A as a regularization instead of directly as part of the network for graph structure-based learning is probably a more rigorous modeling because the incompleteness of the bipartite network might mislead the random walk.

4 Experiments and Discussions

BiRW was compared to CIPHER [5], PRINCE [8] and RWRH [9], three of the best performing algorithms for disease gene prioritization, by 100-fold cross-validation and testing of an independent holdout set with OMIM data. We also compared the three variants of BiRW, BiRW_avg (default for BiRW), BiRW_seq and BiRW_bl, with similar experiments.

4.1 Data Preparation

The disease phenotype network is an undirected graph with 5080 vertices representing OMIM disease phenotypes, and edges weighted in $[0, 1]$. The edge weights measure the similarity between two phenotypes by their overlap in the text and the clinical synopsis in OMIM records, calculated by text mining [10]. The disease-gene associations are represented by an undirected bipartite graph with edges connecting phenotype nodes with their causative gene nodes. Two versions (May-2007 Version and May-2010 Version) of OMIM associations were used in the experiments. May-2007 Version contains 1393 associations between 1126 disease phenotypes and 916 genes, and May-2010 Version contains 2469 associations between 1786 disease phenotypes and 1636 genes. Human protein-protein interaction (PPI) network was obtained from HPRD [12]. The PPI network contains 34,364 curated binary interactions between 8919 genes.

4.2 Comparison with Other Methods

Since the disease gene prioritization algorithms rank genes based on their predicted association against a particular query phenotype, to make a reasonable comparison with CIPHER [5], PRINCE [8] and RWRH [9], the three algorithms were applied to predict the disease genes for each phenotype and the predictions are compared with the results of BiRW phenotype-wise. In the experiment, a disease phenotype was used as a query by an algorithm to rank the genes by their association scores against the query phenotype. For PRINCE and BiRW, the phenotype similarity network was transformed by a logistic function [8]. For all the methods, a 100-fold cross-validation on the OMIM May-2007 Version was performed for parameter tuning, and then the methods were applied to predict the associations in an independent set of associations added into OMIM between May-2007 and May-2010.

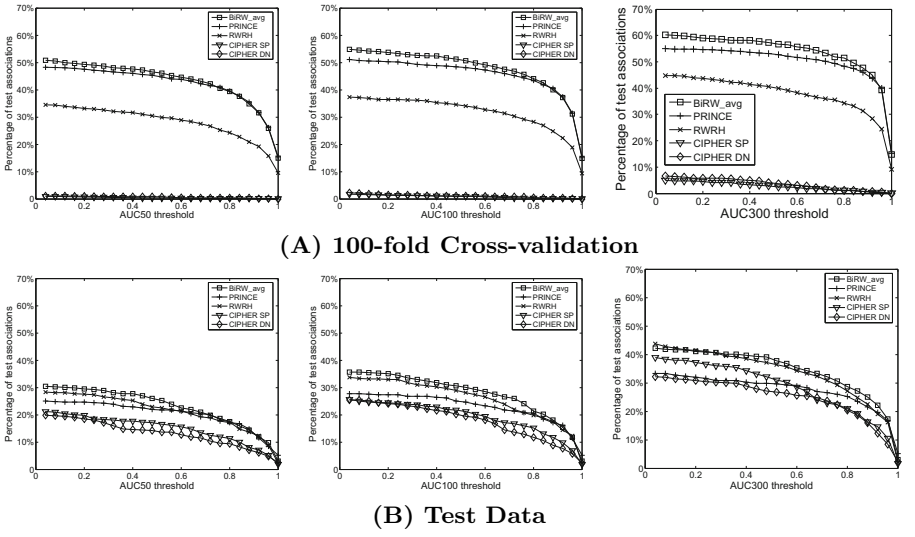


Fig. 3. Performance of predicting OMIM associations. The plots show the percentage of phenotypes, for which a given method achieved a ROC score exceeding a threshold in cross-validation and testing.

There are 1126 disease phenotypes with at least one known causal gene in OMIM version May-2007. In the 100-fold cross-validation, the 1126 disease phenotypes were randomly divided into 100 subsets. In each cross-validation trial, the OMIM associations of the 1% disease phenotypes in a subset were removed, and then used as queries to rank the candidate genes. The hyper-parameters α for both PRINCE and BiRW were chosen from $\{0.1, 0.2, ..., 0.9\}$, and l and r were taken to be between 1 step to 5 steps. The three hyper-parameters of RWRH are set to be the optimal parameters $(0.5, 0.7, 0.5)$ suggested by the experiments in [9]. The test set contains new associations of 518 phenotypes in OMIM May-2010 Version. ROC score (Area Under the Curve of Receiver Operating Characteristic) was used as the global performance measure. The higher the target genes of a query phenotype in the ranking, the better the performance. Specifically, for each phenotype query, the target genes were labeled as positives and the other genes were labeled as negatives. AUCs were computed by the positions of the positives in the ranking list. We reported the AUC with up to 50, 100 and 300 false positives since the top part of AUC is more important.

The results produced by the best parameters in the cross-validation of each method is reported in Fig. 3A ($l = 4$, $r = 4$ and $\alpha = 0.8$ for BiRW and $\alpha = 0.1$ for PRINCE). To make a comprehensive comparison, we plot the number of phenotype queries with a AUC higher than a certain threshold in the plots. The BiRW algorithm performed the best. Out of the 1126 phenotypes, BiRW ranked around 55% in top 50 and 63% in top 500. PRINCE also gave decent prediction performance although BiRW consistently outperformed PRINCE in all the measures. RWRH, CIPHER DN (direct neighbor) and SP (shortest path)

Table 2. Statistical significance in performance comparison. A pairwise comparison by paired t-test of the ranking results in 100-fold cross-validation.

(A) p-values for AUC_{50} comparison

	BiRW(0.8,4,4)	PRINCE(0.1)	RWRH(0.5,0.7,0.5)	C-SP	C-DN
BiRW	NaN				
PRINCE	0.046	NaN			
RWRH	4.41e-037	6.08e-030	NaN		
CIPHER SP	6.97e-158	1.87e-150	1.20e-091	NaN	
CIPHER DN	9.81e-158	7.99e-150	6.87e-090	0.836	NaN

(B) p-values for AUC_{100} comparison

	BiRW(0.8,4,4)	PRINCE(0.1)	RWRH(0.5,0.7,0.5)	C-SP	C-DN
BiRW	NaN				
PRINCE	4.73e-004	NaN			
RWRH	7.30e-039	4.27e-027	NaN		
CIPHER SP	1.65e-175	2.09e-160	1.09e-100	NaN	
CIPHER DN	1.65e-176	1.94e-161	2.71e-099	0.79	NaN

produced inferior results in this experiment. The possible reason for the worse results of CIPHER might be because the associations of the test phenotypes were all removed (called *ab initio* experiment) and each cross-validation held out a significant number of known associations. Thus, no direct neighbors were available for the correlation calculation for many phenotype queries by CHPHER. PRINCE, RWRH and BiRW worked much better than CIPHER SP and CIPHER DN because label propagation and bi-random walk both explore more global information of the networks. We also measured the statistical significance of the difference in AUC_{50} and AUC_{100} by paired t-test. The p-values are reported in Table 2. Clearly, BiRW performs significantly better than all other methods at the significance level 0.05.

4.3 Comparison of BiRW Variants

To understand the effect of combining left walk and right walk with different strategies, we compared BiRW_avg, BiRW_seq and BiRW_bl with the same experiments on OMIM data. The results are reported in Table 3. BiRW_avg and BiRW_seq, which perform random walk with a limited number steps, performed significantly better than BiRW_bl, which performs random walk till the convergence to the stationary distribution. The observation partially agrees with the results by [20] [21], which showed that genes within two-steps are more functional cohesion in the PPI network. When the random-walk steps are above 2 in the gene network, results are very close to optimal as long as the number of steps in the phenotype network is properly chosen. Since the results depends on the random-walks in two networks and the decay factor, we found that it is better to treat the steps as parameters as in BiRW_avg and BiRW_seq. It is also interesting that BiRW_avg performed constantly better than BiRW_seq although the difference is only marginal. We suspect that there might be a bias

Table 3. Comparison of the three BiRW Variants. The table reports a comparison of the ranking results by the BiRW variants, BiRW_avg, BiRW_seq and BiRW_bl. The parameters α, m and n of BiRW are chosen by the 100-fold cross-validation. AUCs up to 50, 100, 300, 500, 1000 and all false positives are reported.

(A) 100-fold Cross-validation

	AUC_{50}	AUC_{100}	AUC_{300}	AUC_{500}	AUC_{1000}	AUC
BiRW_avg(0.8,4,4)	0.4349	0.4818	0.5455	0.5721	0.6097	0.8063
BiRW_seq(0.8,4,3)	0.4295	0.4696	0.5323	0.5596	0.5972	0.8019
BiRW_bl(0.8)	0.2730	0.3344	0.4229	0.4608	0.5138	0.7768

(B) Test Data

	AUC_{50}	AUC_{100}	AUC_{300}	AUC_{500}	AUC_{1000}	AUC
BiRW_avg(0.8,4,4)	0.2321	0.2809	0.3498	0.3862	0.4494	0.7708
BiRW_seq(0.8,4,3)	0.2235	0.2651	0.3344	0.3700	0.4344	0.7672
BiRW_bl(0.8)	0.1675	0.2198	0.3167	0.3689	0.4461	0.7754

in choosing the order of left walk and right walk when BiRW_seq performs sequential random walks, and the bias might be data dependent. In BiRW_avg, there is no ambiguity in the order of the bi-random walk and thus, there might be less variation expected in different data.

5 Conclusion

In the paper, we introduced a bi-random walk algorithm (BiRW) for disease gene prioritization. We analyzed the algorithm by comparison with other random walk algorithms for disease gene prioritization with both algorithmic analysis and empirical experiments. We concluded that BiRW is an effective algorithm for disease gene prioritization and the steps of random walks play a crucial role in the performance of the algorithms. In future, we plan to explore other variations of BiRW to more effectively utilize the hidden information in the networks.

Acknowledgement. This work is supported in part by grant III 1117153 from National Science Foundation.

References

1. Consortium The Wellcome Trust Case Control. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. Nature 447, 661–678 (2007)
2. Johnson, A., O'Donnell, C.: An open access database of genome-wide association resutls. BMC Med. Gent. 10, 6 (2009)
3. Franke, L., Bakel, H., Fokkens, L., et al.: Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. Am. J. Hum. Genet. 78, 1011–1025 (2006)

4. Köhler, S., Bauer, S., Horn, D., et al.: Walking the Interactome for Prioritization of Candidate Disease Genes. Am. J. Hum. Genet. 82, 949–958 (2008)
5. Wu, X.B., Jiang, R., Zhang, M.Q., et al.: Network-based global inference of human disease genes. Mol. Syst. Biol. 4 (2008)
6. Linghu, B., Snitkin, E.S., Hu, Z., et al.: Genome-wide prioritization of disease genes and identification of disease-disease associations from an integrated human functional linkage network. Genome Biol. 10, R91 (2009)
7. Hwang, T.H., Kuang, R.: A Heterogeneous Label Propagation Algorithm for Disease Gene Discovery. In: Proc. of SIAM Intl. Conf. on Data Mining, pp. 583–594 (2010)
8. Vanunu, O., Magger, O., Ruppin, E., et al.: Associating genes and protein complexes with disease via network propagation. PLoS Comput. Biol. 6, e1000641 (2010)
9. Li, Y., Patra, J.C.: Genome-wide inferring gene-phenotype relationship by walking on the heterogeneous network. Bioinformatics 26, 1219–1224 (2010)
10. van Driel, M.A., Bruggeman, J., Vriend, G., et al.: A text-mining analysis of the human phenome. Eur. J. Hum. Genet. 14, 535–542 (2006)
11. McKusick, V.A.: Mendelian inheritance in man and its online version, OMIM. Am. J. Hum. Genet. 80, 588–604 (2007)
12. Peri, S., Navarro, J.D., Amanchy, R., et al.: Development of human protein reference database as an initial platform for approaching systems biology in humans. Genome Res. 13, 2363–2371 (2003)
13. Chuang, H., Lee, E., Liu, Y., et al.: Network-based classification of breast cancer metastasis. Mol. Syst. Biol. 3, 140 (2007)
14. Singh, R., Xu, J., Berger, B.: Pairwise Global Alignment of Protein Interaction Networks by Matching Neighborhood Topology. Res. in Comp. Mol. Biol. 4453, 16–31 (2007)
15. Li, Z., Zhang, S., Wang, Y., et al.: Alignment of molecular networks by integer quadratic programming. Bioinformatics 23, 1631–1639 (2007)
16. Guo, X., Hartemink, A.J.: Domain-oriented edge-based alignment of protein interaction networks. Bioinformatics 25, i240–i246 (2009)
17. Zaslavskiy, M., Bach, F., Vert, J.P.: Global alignment of protein-protein interaction networks by graph matching methods. Bioinformatics 25, i259–i267 (2009)
18. Singh, R., Xu, J., Berger, B.: Global alignment of multiple protein interaction networks with application to functional orthology detection. Proc. Natl. Acad. Sci. U.S.A. 105, 12763–12768 (2008)
19. Zhou, D., et al.: Learning with Local and Global Consistency. Advanced Neural Information Processing Systems 16, 321–328 (2004)
20. Chua, H., Sung, W., Wong, L.: Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions. Bioinformatics 22, 1623–1630 (2006)
21. Xu, J., Li, Y.: Discovering disease-genes by topological features in human protein-protein interaction network. Bioinformatics 22, 2800–2805 (2006)

Selecting Feature Subset via Constraint Association Rules*

Guangtao Wang and Qinbao Song

Dept. of Computer Science and Technology
Xi'an Jiaotong University, China

Abstract. In this paper, a novel feature selection algorithm FEAST is proposed based on association rule mining. The proposed algorithm first mines association rules from a data set; then, it identifies the relevant and interactive feature values with the constraint association rules whose consequent is the target concept, and detects the redundant feature values with constraint association rules whose consequent and antecedent are both single feature value. After that, it eliminates the redundant feature values, and obtains the feature subset by mapping the relevant feature values to corresponding features. The efficiency and effectiveness of FEAST are tested upon both synthetic and real world data sets, and the classification results of the three different types of classifiers (including Naive Bayes, C4.5 and PART) with the other four representative feature subset selection algorithms (including CFS, FCBF, INTERACT and associative-based FSBAR) were compared. The results on synthetic data sets show that FEAST can effectively identify irrelevant and redundant features while reserving interactive ones. The results on the real world data sets show that FEAST outperformed other feature subset selection algorithms in terms of average classification accuracy and Win/Draw/Loss record.

Keywords: Feature subset selection, association rule, feature interaction.

1 Introduction

Feature subset selection is an important research issue in the domains of machine learning and data mining. Its purpose is to help the learning algorithm focus on those aspects of the data most useful for analysis and future prediction. Generally, feature subset selection is the process of identifying and removing as many irrelevant and redundant features as possible. As irrelevant features do not contribute to the predictive accuracy [13], and redundant features do not contribute to getting a better predictor for that the most information they provide is already present in other feature(s) [28], thus many feature subset

* This work is supported by the National Natural Science Foundation of China under grant 61070006.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 304–321, 2012.

selection algorithms have been proposed to handle the irrelevant features or/and redundant features.

However, feature interaction is not a negligible issue in practice [12]. For example, suppose $F_1 \oplus F_2 = Y$, where F_1 and F_2 are two boolean variables, Y represents the target concept, and $\oplus$ represents the *xor* operation. F_1 and F_2 are irrelevant with Y when we consider their discrimination abilities for Y separately, but they become very relevant when we combine them together. Therefore, removing the interactive features will lead to poor predictive accuracy. Thus a feature subset selection algorithm should consist of eliminating the irrelevant and redundant features while taking the feature interaction into consideration. Unfortunately, to our knowledge, only a few algorithms can deal with this situation [12,29].

Association rule mining can discover interesting associations among data items [15], it has been used to build classifiers which show better classification accuracy compared with the other types of classifiers [2,10,19]. Especially, it also has been employed for feature selection recently by Xie et al. [26]. However, Xie et al. only focus on relevant features and do not consider redundant and interactive features.

An association rule is an expression of $A \Rightarrow C$, where A (Antecedent) and C (Consequent) are itemsets. If we view A as the feature(s) and C as the feature(s)/the target concept, association rules can reveal the dependencies between either feature(s) and feature(s) or feature(s) and the target concept. Therefore, it is reasonable and desirable to devise an association rule mining based method to choose feature subset.

In this paper, we propose a Feature subset sElection Algorithm based on aSsociaTion rule mining (FEAST), which can eliminate the irrelevant and redundant features while taking the feature interaction into consideration. Moreover, FEAST uses association as the measure to evaluate the relativity between feature(s) and the target concept, which is quite different from the traditional measures, such as the consistency measure [4,20,29], the dependence measure [9,27], the distance measure [18,21] and the information theory measure [17,23]. The association measure evaluates irrelevant, redundant and interactive features in a uniform way, it is at least a potential alternatives for feature subset selection. The experimental results on the synthetic and real world data sets show the effectiveness of the proposed algorithm.

The rest of the paper is organized as follows: In Section 2, we introduce the related work. In Section 3 we describe some preliminaries. In Section 4, we present the new feature subset selection algorithm FEAST. In Section 5, we provide the experimental results. Finally, in Section 6, we summarize our work and draw some conclusions.

2 Related Work

Feature subset selection has been an active research topic since 1970's, and a great deal of research has been published.

Of the existing research work, most feature selection algorithms can effectively identify the irrelevant features based on different evaluation functions. But not all of them can eliminate the redundant features and take the feature interaction into consideration [3]. Thus, the existing feature selection algorithms can generally be grouped into several categories according to whether or not they can deal with irrelevant features, redundant features and the feature interaction.

Traditionally, feature subset selection research has focused on searching for relevant features. Feature weighting/ranking algorithms [8] weigh features individually and rank them based on their relevance to the target concept. Unfortunately, they are incapable of removing redundant features. Such as well-known Relief and its extension Relief-F [18].

Moreover, along with irrelevant features, redundant features also affect the speed and accuracy of learning algorithms and thus should be eliminated as well [16]. CFS [9], FCBF [27] and CMIM [5] are examples that take into consideration the redundant features. However, they do not handle the feature interaction [29].

Feature interaction has been drawing more attention in recent years. There can be two-way, three-way or complex multi-way interactions among features [7]. Jakulin and Bratko [12] use interaction gain as a heuristic to detect feature interaction. Their algorithms can detect 2-way (one feature and the class) and 3-way (two features and the class) interactions. Zhao and Liu [29] demonstrate that feature interactions can be implicitly handled by a carefully designed feature evaluation metric and a search strategy with a specially designed data structure.

Recently, association rules have been used for feature selection. Xie et al. [26] propose an association rule-based feature selection algorithm FSBAR. Unfortunately, it just detects relevant features and does not handle redundant and interactive features. In contrast, our algorithm aims to eliminate the irrelevant and redundant features, and takes the multi-way feature interactions into consideration, hence it is quite different from these algorithms above.

3 Preliminaries

3.1 Strong, Classification and Atomic Association Rules

Association rule mining searches for interesting relationships among items in a data set D. Let $I = \{i_1, i_2, \cdots, i_k\}$ be a set of items, an association rule is an implication of form $A \Rightarrow B$, where $A \subset I$, $B \subset I$, and $A \cap B = \phi$.

The support and confidence are two important measures of a rule's interestingness.

1. The support of rule $A \Rightarrow B$ is the percentage of instances in D that contain both A and B, denoted as $\text{Support}(A \Rightarrow B) = P(A \cup B)$; this measure reflects the rule's usefulness whose value range is $(0, 100\%]$.
2. The confidence of rule $A \Rightarrow B$ is the percentage value that shows how frequently B occurs among all the instances containing A. It is denoted as $\text{Confidence}(A \Rightarrow B) = P(B|A)$; this measure reflects the rule's certainty whose value range is $(0, 100\%]$.

Typically, association rules are considered interesting if they satisfy minimum support threshold (*minSupp*) and minimum confidence threshold (*minConf*). The *minSupp* and *minConf* can be set by users or domain experts. Based on these two thresholds, strong association rule (SAR) can be defined as follow.

Definition 1. *Strong association rule (*SAR*). A rule r of form $A \Rightarrow C$ is a strong association rule if and only if:*

$$Supp(r) > minSupp \wedge Conf(r) > minConf. \tag{1}$$

Where *Supp(r)* and *Conf(r)* represent the support and confidence of the association rule *r*, respectively.

For the sake of introducing classification association rule (CAR) and atomic association rule (AAR), we first give the concepts of feature value itemset (FVIS) and target value itemset (TVIS).

Let $D = \{d_1, d_2, \cdots, d_n\}$ be a data set of n instances, $F = \{F_1, F_2, \cdots, F_m\}$ be the feature space of D with m features, where F_i is the domain of ith feature and Y be the target concept. The instance d_i of D can be denoted as a tuple (X_i, y_i), where $X_i \in F_1 \times F_2 \times \cdots \times F_m$, and $y_i \in Y$. Then the feature value itemset $\text{FVIS} = \bigcup_{i=1}^{m} F_i$ containing all possible feature values, and the target value item set $\text{TVIS} = Y$.

With the definitions of FVIS and TVIS, classification association rule (CAR) and atomic association rule (AAR) are defined as follows.

Definition 2. *Classification association rule (*CAR*). A rule r of form $A \Rightarrow C$ is a classification association rule if and only if:*

$$r \text{ is a SAR } \wedge A \subseteq \text{FVIS } \wedge C \subseteq \text{TVIS} \wedge \mid C \mid = 1. \tag{2}$$

Here, $|X|$ denotes the cardinality of set X. All CARs constitute *classification association rule set* (CARset).

Definition 3. *Atomic association rule (*AAR*). A rule r of form $A \Rightarrow C$ is an atomic association rule if and only if:*

$$r \text{ is a SAR} \wedge \mid A \mid = 1 \wedge \mid C \mid = 1. \tag{3}$$

All AARs excluding atomic classification rules constitute *atomic association rule set* (AARset). Here, an atomic classification rule is an AAR whose consequent is the target concept value.

3.2 Definitions of Relevant, Redundant and Interactive Features

To define the relevant, redundant features and feature interaction based on constraint association rules (i.e., classification and atomic association rules), we firstly give the definitions of relevant feature value, redundant feature value and feature value interaction based on association rules.

Definition 4. *Relevant feature value (*RelFV*). A specific value f_{ij} of feature F_i is relevant to the target concept Y if and only if:*

$$\exists r \in \text{CARset}, f_{ij} \in r.Ante. \tag{4}$$

*Otherwise, f_{ij} is an irrelevant feature value (*iRelFV*).*

Where f_{ij} denotes the jth $(1 \leq j \leq |F_i|)$ value of feature F_i, and $r.Ante$ represents the antecedent of rule r. The same notations are employed in the following definitions.

From Definition 4 we can know that, the feature values appeared in the antecedent of a rule $r \in$ CARset are relevant feature values; on the other hand, the feature values never appeared in the antecedent of any rule $r \in$ CARset are irrelevant feature values.

We know that classification association rules have been extensively employed in classification [2,10,19], and these classifiers usually possess preferable classification accuracy. This indicates that the rules in CARset can be used to effectively explore the relationship between features and target concept. The feature values appeared in the antecedents of CARs are necessary and related to the target concept. Thus, it is reasonable to identify the relevant feature values by Definition 4.

However, the feature values appeared in a rule's antecedent maybe redundant. That is, two closely-correlated feature values will be simultaneously appearing in the rule's antecedent. This is because that the association rules are generated based on frequent itemset mining (FIM) [24], but FIM cannot detect the redundant items (i.e., feature values) since that, for a given feature value, if it is frequent and selected into a frequent itemset, then the value being redundant to it will be frequent and selected into an itemset as well. To handle this problem, the redundant feature value is defined as follow.

Definition 5. *Redundant feature value (*RedFV*). A specific value f of a feature value set (*FVset*) is redundant if and only if:*

$$\exists r \in \text{AARset}, (\{f\} = r.Cons) \wedge (r.Ante \subseteq \text{FVset}). \tag{5}$$

Where $r.Ante$ and $r.Cons$ represent the antecedent and consequent of rule r, respectively.

From Definition 5 we can know that, of a given feature value set, a feature value is redundant when it appeared in the consequent of a rule in AARset and the rule's antecedent is in the given feature value set as well.

As we known, for a redundant feature value, the information it provides is already present in other feature value. This indicates that it is closely related to and can be replaced by other feature value. What's more, atomic association rule can be used to explore the correlation between two feature values. Thus, Definition 5 based on AAR can be used to detect redundant value.

It is noticed that Definition 5 only shows the two-way value redundancy (the redundancy between two values). Of course, there might exist multi-way feature

value redundancy (the redundancy among multiple feature values). However, detecting all the multi-way value redundancy is a combination explosion problem since we need to list all possible combinations. This is impracticable even when the feature space is of a middle size. Therefore, we just focus on the two-way redundancy in this paper.

Suppose FVset $= \{f_1, f_2, \cdots, f_k\}$ is a feature value set with k feature values. It is a value-assignment set of a feature set Fset with k features, that is, each member of FVset corresponds to exactly a value of the feature of Fset. Let ($A \subset$ FVset) $\neq \phi$ and $B =$ FVset $- A$, y be a value of the target Y, $Conf(r)$ be the confidence of an association rule r, and r_F, r_A and r_B be the CARs of FVset $\Rightarrow \{Y = y\}$, $A \Rightarrow \{Y = y\}$ and $B \Rightarrow \{Y = y\}$, respectively. Then, the interactive feature value can be defined as follow.

Definition 6. *k-th feature value interaction. The k feature values in* FVset *are said to interact with each other if and only if:*

$$Conf(r_F) > Conf(r_A) \ \wedge \ Conf(r_F) > Conf(r_B). \tag{6}$$

The confidence of an association rule shows how well the rule's antecedent describes its consequent. The higher confidence means the stronger description ability. In Definition 6, the confidence of rule r_F is greater than those of rules r_A and r_B. This means that although either feature value set A or B is not helpful in describing the target concept, FVset $= A \cup B$ works well in describing the target concept. In this case, feature value sets A and B are said to interact with each other.

According to Definition 2, the classification association rules usually have high confidence since their confidence should be at least greater than *minConf*. This implies that all the rules with high confidence are included in CARset. In Definition 6, it is impossible that r_A or r_B is a CAR but r_F is not a CAR, since $Conf(r_F)$ is greater than both $Conf(r_A)$ and $Conf(r_B)$. Therefore, the antecedents of rules in CARset will contain all possible feature value interactions according to Definition 6. That is, the feature value interaction can be reserved by the rules in CARset.

Based on the definitions of relevant feature value (RelFV), redundant feature value (RedFV) and feature value interaction, relevant feature, redundant feature and feature interaction are defined as follows.

Definition 7. *Relevant feature (*RelFea*). Feature F_i is relevant to the target concept Y if and only if:*

$$\exists f_{ij} \in F_i, \{f_{ij} \mid f_{ij} \textit{ is a } \text{RelFV}\} \neq \phi. \tag{7}$$

*Otherwise, F_i is an irrelevant feature (*iRelFea*).*

Definition 7 shows that a feature is relevant when at least one of its values is a relevant feature value. On the other hand, for an irrelevant feature, all its values are irrelevant.

Definition 8. *Redundant Feature (*RedFea*). Feature F_i is redundant if and only if:*

$$\forall f_{ij} \in F_i, \{f_{ij} \mid f_{ij} \textit{ is a } \text{RedFV } \textit{or an } \text{iRelFV}\} \neq \phi. \tag{8}$$

Definition 8 indicates that a feature is redundant due to two reasons: (i) each value of this feature is a redundant feature value; (ii) some values of this feature are redundant while others are irrelevant. As irrelevant values provide no information about the target concept and redundant values provide the information which is present by the other values, they are all useless in describing the target concept. This is consistent with the property of the classical definition of redundant feature [28].

Definition 9. *Feature interaction. Let* Fset $= \{F_1, F_2, \cdots, F_k\}$ *be a feature subset with k features, and* VAset *be its value-assignment sets. Features* $F_1, F_2, \cdots, F_k$ *are said to interact with each other if and only if:*

$$\exists \text{fset} \in \text{VAset}, \{\text{fset } \textit{is a } \text{FVset } \textit{with k-th feature value interaction}\} \neq \phi. \tag{9}$$

As we known, there is an intrinsic relationship between a feature and its values, and the properties of a feature subset can be studied by its value-assignment. Thus, for a given feature subset, it is reasonable that the feature interaction among this feature subset could be implied and studied by that among its value-assignment. Inspired by this, Definition 9 based on feature value interaction is proposed to identify feature interaction.

4 Feature Subset Selection Algorithm

Based on the definitions of relevant feature, redundant feature and feature interaction, we propose a novel feature subset selection algorithm FEAST, which searches for relevant features while taking into consideration redundant features and feature interaction.

4.1 FEAST Algorithm

The algorithm FEAST consists of four steps: i) *Association rule mining*, ii) *Relevant feature value set discovery*, iii) *Redundant feature value elimination* and iv) *Feature subset identification*.

1) *Association rule mining*
   Constraint association rules are mined from the given data set based on the predetermined thresholds *minSupp* and *minConf*. These rules include classification association rules and atomic association rules. After this step, classification association rule set (CARset) and atomic association rule set (AARset) are obtained.
2) *Relevant feature value set discovery*
   By collecting the antecedents of rules in CARset together, initial relevant feature value set (RFVset), which reserves the feature value interactions, is achieved according to Definition 4 and Definition 6.

3) *Redundant feature value elimination*
A feature value is redundant means that the information it provides is already present in another feature value. This indicates the redundant value is implied by another value. In this paper, atomic association rule is employed to identify this kind of implication relation. The higher the confidence of an atomic association rule is, the stronger the implication. This means that the AARs with higher confidence could be used to identify and eliminate redundant values firstly.

For a given AAR $r \in$ AARset with the highest confidence, the feature value in r's consequent is identified redundant and eliminated from current RFVset. Meanwhile, according to Definition 5, a feature value in the consequent of an AAR is redundant iff the feature value of the AAR's antecedent is in the current RFVset. Therefore, after eliminating r's consequent from RFVset, AARset should be updated by removing r and the rules whose antecedents are equal to r's consequent.

4) *Feature subset identification*
After eliminating redundant feature values, there are no irrelevant and redundant values in RFVset. Meanwhile, step 2 shows that RFVset includes all feature value interactions based on which the feature interactions are defined (see details in Definition 9). Thus, according to Definition 7, by mapping the feature values in RFVset to the corresponding features, the final feature subset is identified, which not only retains relevant features and excludes irrelevant and redundant features, but also takes feature interaction into consideration.

Algorithm 1 shows the pseudo-code description of FEAST. Of the input parameters, *minSupp* and *minConf* are used as the constraint conditions to achieve strong association rule SAR (Definition 1).

The pseudo-code of FEAST includes four parts, in part 1 (lines 1-2), classification association rule set CARset and atomic association rule set AARset are mined by function *FP_growth* [11] on the given data set D according to *minSupp* and *minConf*. In part 2 (lines 3-4), the union of the antecedents of the association rules in CARset constitutes the relevant feature value set RFVset. Part 3 (lines 5-13) is used to eliminate the redundant feature values in RFVset, where function *Sort* sorts the rules in AARset in descending order of rule's confidence. Firstly, the first rule (i.e. the rule with the highest confidence) r is chosen and removed from AARset. Then if its antecedent is a subset of the current RFVset, the value in r's consequent is eliminated from RFVset; meanwhile, the rules whose antecedents are equal to its consequent are removed from AARset. This process repeats until that AARset is empty. Part 4 (lines 14-17) achieves the selected feature subset S according to the feature values in RFVset.

Time Complexity Analysis. In part 1, the CARset and AARset are mined by function *FP_growth*. Since the time consumption of FP-growth is closely related to the value of *minSupp* [11], the time complexity of this part can be represented as *O(f(minSupp, D))*, where *f(minSupp, D)* is a function of *minSupp* and D which increases with the decrease of *minSupp*/increase of the size of D. For part

Algorithm 1. FEAST

```
inputs : D - the given data set;
         minSupp - the support threshold;
         minConf - the confidence threshold.
output: S - selected feature subset.
//– Part 1 : Association rule mining –
S = φ; RFVset = φ;//RFVset- relevant feature value set;
[CARset, AARset ] = FP_growth (D, minSupp, minConf);
//– Part 2 : Relevant feature value set discovery –
for each r ∈ CARset do
    RFVset = RFVset ∪ r.Antecedent;
//– Part 3: Redundant feature value elimination –
Sort (AARset); //sort rules in descending order of rule's confidence
while AARset ≠ φ do
    r = the first rule in AARset;
    AARset = AARset − {r};
    if r.Antecedent ⊂ RFVset then
        RFVset = RFVset − r.Consequent;
        for each r′ ∈ AARset do
            if r′.Antecedent == r.Consequent then
                AARset = AARset − {r′};
//– Part 4: Feature subset identification –
for each feature value val ∈ RFVset do
    if val ∈ value set of feature F then
        S = S ∪ {F};
return S
```

2, once a CAR is generated by FP-growth, its antecedent could be merged into RFVset meanwhile, so the consumed time of this part can be ignored. For part 3, since its main time consummation is the process of sorting the rules in AARset, the time complexity of this part is $O(V \cdot \log V)$ (by quick sort), where V is the number of rules in AARset. The time complexity of part 4 is $O(K)$ where K is the number of feature values in the final RFVset whose maximum value is the number of all possible feature values in D.

Consequently, the time complexity of FEAST is $O(f(minSupp, D) + O(V \cdot \log V) + O(K)$. Since part 1 is the major time consumer in the worst case, the efficiency of FEAST depends largely on that of association rule mining.

5 Experimental Results and Analysis

In this section, we empirically evaluate the performance of FEAST, and present the experimental results compared with the other four representative feature selection algorithms upon both synthetic and real world data sets.

5.1 Benchmark Data Sets

Synthetic Data Sets. In order to directly evaluate how well FEAST deals with irrelevant, redundant features and feature interaction, five synthetic data sets with all the irrelevant, redundant and interactive features being known are employed.

The first two data sets synData1 and synData2 are generated by the data generation tool RDG1 of the data mining toolkit WEKA[1]. The other three data sets about MONK's problems are available from UCI Machine Learning Repository [1]. The five data sets are described as follows.

1) synData1. There are 100 instances and 10 boolean features $a_0, a_1, \cdots, a_9$. The target concept c is defined by $c = (a_0 \wedge a_1 \wedge \overline{a_5}) \vee (a_0 \wedge \overline{a_1} \wedge a_6 \wedge a_8) \vee (a_0 \wedge a_1 \wedge a_5 \wedge a_8) \vee (\overline{a_0} \wedge a_1 \wedge a_5 \wedge \overline{a_8}) \vee (a_5 \wedge a_6 \wedge a_8) \vee (a_0 \wedge \overline{a_1})$.
2) synData2. There are 100 instances, 11 boolean features denoted as $a_0, a_1, \cdots, a_9$ and a redundant feature r that is the copy of a_5. The target concept c is defined by $c = \overline{a_5} \vee (\overline{a_1} \wedge \overline{a_6} \wedge \overline{a_8})$.
3) MONK1. There are 432 instances and 6 features $a_1, a_2, \cdots, a_6$. The target concept c is defined by $c = (a_1 = a_2) \vee (a_5 = 1)$.
4) MONK2. There are 432 instances and 6 features $a_1, a_2, \cdots, a_6$. The target concept c is defined by exactly two of $\{a_1 = 1, a_2 = 1, \cdots, a_6 = 1\}$.
5) MONK3. There are 432 instances and 6 features $a_1, a_2, \cdots, a_6$. The target concept c is defined by $c = (a_5 = 3 \wedge a_4 = 1) \vee (a_5 \neq 4 \wedge a_2 \neq 3)$. 5% class noise was added to the training set.

For each data set, the features appearing in the definition of the target concept are all relevant, while the absent features are either redundant or irrelevant. The conjunctive terms in the target concept's definition imply feature interactions.

Real World Data Sets. 14 extensively used real world data sets, which are available from from UC Irvine Machine Learning Repository [1], are employed. Table 1 summarizes the 14 data sets in terms of number of features (denoted as F), the number of instances (denoted as I), the number of target concept values (denoted as T). The sizes of data sets vary from 57 to 20,000 instances, and the total number of original features is up to 240. Note that for the data sets containing continuous-value features, if needed, we apply the MDL discretization method (available in WEKA).

Table 1. Summary of the 14 real world data sets

Data set	F	I	T	Data set	F	I	T
heart-c	11	303	5	autos	22	205	7
cleve	12	303	2	mushroom	22	8124	2
austra	14	690	2	colic-orig	23	368	2
labor	14	57	2	flags	26	194	6
letter	15	20000	26	molecular	57	106	2
primary-tumor	17	339	22	splice	60	3190	3
lymph	18	148	4	mfeat-pixel	240	2000	10

5.2 Experimental Setup

1) Four representative feature selection algorithms were selected to be compared with FEAST.

These algorithms include two well-known and frequently-used CFS [9] and FCBF [27]. They can effectively identify irrelevant features while taking consideration of the redundant features.

[1] http://www.cs.waikato.ac.nz/ml/weka/

To further study the performance of FEAST in terms of handling feature interaction, an algorithm INTERACT [29], which is specifically proposed to address the feature interaction, is selected as one benchmark algorithm.

Moreover, since our proposed FEAST is an association-rule-based feature selection algorithm, a latest association-rule-based feature selection algorithm FSBAR [26] is selected as well.

The parameters of these algorithms (including FEAST) were determined by the cross-validation strategy.

2) Classification accuracy over selected feature subset is extensively used as a measure to evaluate the performance of the feature selection algorithm in feature selection literature. This is due to the fact that the relevant features of real world data sets are usually not known in advance, and we can not directly evaluate how good a feature selection algorithm is by the features selected.

However, different classification algorithms have different biases, and a feature subset selection algorithm may be more suitable for some classification algorithms than others. With this in mind, three different types of well-known classification algorithms including probability-based Naive Bayes [14], decision tree-based C4.5 [22] and rule-based PART [6] were selected.

In order to make best use of the data set and get stable results, the classification accuracies before and after feature selection were obtained by a 5×10-fold cross-validation procedure. That is, for a given data set, each feature selection algorithm and each classifier were repeatedly performed on the data set with 10-fold cross-validation by five times.

3) All the experiments were conducted in the WEKA environment [25].

5.3 Results on the Synthetic Data Sets

Table 2 shows the feature subsets selected by the five feature subset selection algorithms on the five synthetic data sets. In this table, '_' indicates a missing relevant feature, and the letter in bold type indicates an irrelevant or a redundant feature selected by mistake. The last row "Relevant features" reports the actual relevant features of each data set.

Table 2. Features selected by the five algorithms on the synthetic data sets

FSS algorithm	synData1	synData2	MONK1	MONK2	MONK3
CFS	$a_0, \_, a_5, a_6, a_8$	$\boldsymbol{a_0}, a_1, a_5, \_, \boldsymbol{a_7}, \_, \boldsymbol{r}$	$\_, \_, a_5$	$\_, \_, \_, \_, a_5, \_$	$a_2, \_, \_$
FCBF	$a_0, \_, a_5, a_6, a_8$	$\boldsymbol{a_0}, a_1, a_5, \_, \boldsymbol{a_7}, \_$	$\_, \_, a_5$	$a_1, a_2, a_3, a_4, a_5, a_6$	a_2, a_4, a_5
FSBAR	$a_0, a_1, \boldsymbol{a_3}, a_5, a_6, a_8$	$\boldsymbol{a_0}, a_1, a_5, a_6, a_8$	$\_, \_, a_5$	$a_1, \_, \_, \_, \_, \_$	$a_2, \_, a_5$
INTERACT	a_0, a_1, a_5, a_6, a_8	$a_1, \boldsymbol{a_3}, \boldsymbol{a_4}, a_5, a_6, \boldsymbol{a_7}, \_$	a_1, a_2, a_5	$a_1, a_2, a_3, a_4, a_5, a_6$	a_2, a_4, a_5
FEAST	a_0, a_1, a_5, a_6, a_8	a_1, a_5, a_6, a_8	a_1, a_2, a_5	$a_1, a_2, a_3, a_4, a_5, a_6$	a_2, a_4, a_5
Relevant features	a_0, a_1, a_5, a_6, a_8	a_1, a_5, a_6, a_8	a_1, a_2, a_5	$a_1, a_2, a_3, a_4, a_5, a_6$	a_2, a_4, a_5

From Table 2, we observe that: (i) Only algorithm FEAST removes all irrelevant features while reserving all relevant features for all the five data sets. The other algorithms identify the irrelevant on some but not all data sets. (ii) Except algorithm CFS, all other four algorithms can identify and remove the redundant feature r in the data set "synData2". (iii) Only algorithm FEAST

reserves all the interactive features on all the five data sets. INTERACT works well on all the data sets except for "synData2". The other algorithms identify all the interactive features on some but not all the data sets.

5.4 Results on the Real World Data Sets

In this section, we present the comparison results of FEAST with other feature subset selection algorithms in terms of (i) the classification accuracies after feature subset selection; (ii) the proportion of selected features; and (iii) the runtime.

Here, the proportion of selected features is the ratio of the number of features selected by a feature selection algorithm to the original number of features of a data set.

What's more, we also provide the sensitivity analysis results of the support and confidence thresholds on the proposed algorithm FEAST.

Classification Accuracy Comparison. Table 3 records the classification accuracies of Naive Bayes, C4.5 and PART with the five feature subset selection algorithms, and the Win/Draw/Loss records, which are the numbers of data sets where the classification accuracy of the given classifier obtained with FEAST is greater than/equal to/lower than that with the compared feature selection algorithm.

Table 3. Accuracies of Naive Bayes, C4.5 and PART with different feature selection algorithms

Data Set	Naive Bayes						C4.5						PART					
	FEAST	CFS	FCBF	INTERACT	FSBAR	ORG	FEAST	CFS	FCBF	INTERACT	FSBAR	ORG	FEAST	CFS	FCBF	INTERACT	FSBAR	ORG
heart-c	83.46	84.43	84.43	82.90	82.18	84.44	79.80	79.80	79.80	78.88	80.86	78.79	82.13	81.12	81.12	79.80	82.84	78.85
cleve	84.22	84.86	84.86	83.50	82.51	83.85	79.60	79.20	79.20	78.75	78.22	77.90	83.19	80.58	80.58	81.72	78.88	79.57
austra	87.68	85.51	87.10	87.48	86.52	85.22	86.46	85.51	86.52	86.46	87.10	86.70	86.38	85.51	85.07	86.00	86.23	85.80
labor	90.00	89.33	89.33	90.18	89.47	91.67	84.33	80.67	80.67	91.58	85.96	73.68	88.00	80.67	84.00	85.96	85.96	80.67
letter	74.48	73.03	74.48	74.55	NA	74.04	78.98	79.17	79.14	79.08	NA	78.82	81.45	81.41	80.90	81.05	NA	80.69
primary-tumor	47.48	45.70	46.00	49.68	43.95	50.13	43.65	41.56	42.47	41.12	42.18	41.00	43.35	45.39	40.12	40.53	43.07	40.70
lymph	83.62	81.67	80.24	83.24	83.78	83.67	77.62	75.71	70.81	73.51	74.32	78.33	81.71	77.14	78.90	76.08	75.00	79.67
autos	77.95	77.40	69.21	78.15	59.51	71.64	77.98	75.55	67.31	76.98	73.17	83.81	78.95	79.45	67.29	75.90	74.63	78.00
mushroom	95.59	98.52	98.52	98.92	98.92	95.83	100.00	98.52	99.02	100.00	100.00	100.00	100.00	98.52	99.02	100.00	100.00	100.00
colic-orig	83.95	81.52	84.25	70.22	83.15	70.40	85.84	81.52	81.52	66.30	85.33	85.03	85.84	81.52	81.24	66.30	84.24	64.11
flags	79.89	73.13	75.18	70.82	70.1	73.21	71.74	72.24	71.63	70.72	69.59	71.18	70.16	70.58	72.1	66.19	67.53	64.92
molecular	97.18	93.27	95.27	94.53	92.45	90.27	80.91	83.82	82.82	81.70	83.96	80.82	85.82	86.73	84.82	86.98	86.79	82.82
splice	96.24	92.48	96.14	96.13	91.85	95.36	94.54	92.70	94.48	94.31	92.95	94.36	92.76	92.07	93.39	92.93	92.57	92.51
mfeat-pixel	90.95	93.00	91.15	90.45	NA	93.30	77.40	79.60	77.80	80.20	NA	78.65	83.00	84.15	80.95	82.25	NA	82.00
Average	83.76	82.42	82.58	82.20	80.37	81.64	79.92	78.97	78.08	78.54	79.47	79.22	81.62	80.35	79.25	78.69	79.81	77.88
W/D/L	-	10/0/4	8/1/5	9/0/5	10/0/2	8/0/6	-	9/1/4	9/1/4	8/2/4	7/1/4	9/1/4	-	9/0/5	12/0/2	11/0/2	9/1/2	13/1/0

* In this table, "ORG" denotes original data sets, "W/D/L" represents "Win/Draw/Loss", and "NA" means the algorithm is not available.

From Table 3 we observe that:

1) For Naive Bayes, (i) compared to the original data set, the average accuracy of Naive Bayes is improved by all the algorithms except FSBAR; (ii) FEAST outperforms other algorithms in terms of average accuracy, it improves the average accuracy by 2.29% averagely; (iii) FEAST outperforms other algorithms in terms of Win/Draw/Loss record, it wins other algorithms for 9.25 out of 14 data sets on average, while losses only 4 out of 14 on average.
2) For C4.5, (i) compared to the original data set, the average accuracy of C4.5 is improved only by the FEAST and FSBAR, but FSBAR were not available on two data sets due to its high time complexity; (ii) FEAST outperforms other algorithms in terms of average accuracy, it improves the average accuracy

by 1.47% averagely; (iii) FEAST outperforms other algorithms in terms of Win/Draw/Loss record, it wins other algorithms for 8.25 out of 14 data sets on average, while losses only 4 out of 14 on average.

3) For PART, (i) compared to the original data set, the average accuracy of PART is improved by all algorithms; (ii) FEAST outperforms other algorithms in terms of average accuracy, it improves the average accuracy by 2.64% averagely; (iii) FEAST outperforms other algorithms in terms of Win/Draw/Loss record, it wins other algorithms for 10.25 out of 14 data sets on average, while losses only 2.75 out of 14 on average.

Table 4. Proportion (%) of selected features for different feature selection algorithms

Data set	FEAST	CFS	FCBF	INTERACT	FSBAR
heart-c	81.82	54.55	54.55	90.91	90.91
cleve	83.33	50.00	50.00	83.33	83.33
austra	50.00	50.00	50.00	92.86	64.29
labor	57.14	50.00	42.86	50.00	50.00
letter	80.00	73.33	73.33	80.00	NA
primary-tumor	47.06	70.59	64.71	94.12	52.94
lymph	44.44	55.56	44.44	55.56	55.56
autos	27.27	22.73	18.18	27.27	54.55
mushroom	36.36	18.18	18.18	27.27	36.36
colic-orig	26.09	8.70	8.70	21.74	30.43
flags	26.92	11.54	15.38	38.46	57.69
molecular	22.81	10.53	10.53	10.53	12.28
splice	31.67	10.00	36.67	38.33	11.67
mfeat-pixel	48.75	42.92	11.25	14.58	NA
Average	47.40	37.76	35.63	51.78	50.00

Table 5. Runtime (ms) for different feature selection algorithms

Data set	FEAST	CFS	FCBF	INTERACT	FSBAR
heart-c	20	22	20	144	215
cleve	20	76	72	63	412
austra	45	80	83	74	1432
labor	16	62	60	62	18
letter	1190	678	558	5333	NA
primary-tumor	624	74	81	64	228
lymph	51	74	69	89	6786
autos	2216	78	72	82	29206
mushroom	223	238	215	405	242803
colic-orig	31	74	81	88	582
flags	319	22	42	58	2649
molecular	79	82	77	66	631
splice	1890	126	42	889	57435
mfeat-pixel	4250	7287	1696	4514	NA
Average	783.86	640.93	226.29	852.21	28533.08

Proportion of Selected Features Comparison. The reduction on the number of features is an important metric used to evaluate feature subset selection algorithms. This can be measured through the proportion of features selected by the feature selection algorithms.

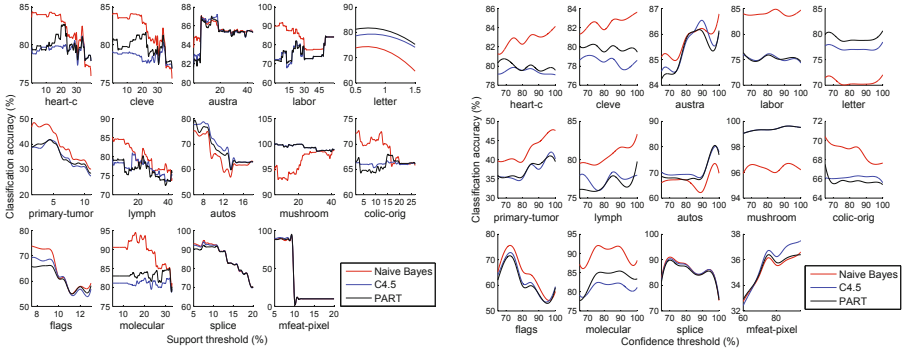
Table 4 presents the proportion of features selected by each of the five feature selection algorithms over the 14 data sets. From this table we observe that: i) All the feature subset selection algorithms could significantly reduce the number of features on average. FCBF ranks 1 with proportion of selected features 35.63%, and INTERACT ranks last with 51.78%. ii) FEAST outperforms algorithms INTERACT and FSBAR in reducing the number of features.

Runtime Comparison. Table 5 records the runtime of each feature subset selection algorithm upon the 14 data sets. From it we observe that (i) the average runtime of different algorithms is varying greatly, FCBF ranks 1 with 226.29 ms, and FSBAR ranks last with 28533.08 ms. (ii) FEAST is faster than INTERACT and FSBAR. Compared with the associative-based algorithm FSBAR, FEAST is much more efficient since it generates association rules by FP-growth algorithm which is more efficient than the Apriori algorithm used in FSBAR.

To summarize, the proposed algorithm FEAST outperformed other feature subset selection algorithms on the 14 UCI data sets in terms of average classification accuracy and Win/Draw/Loss record, and the runtime and the reduction rate are acceptable.

Sensitivity Analysis of the Support and Confidence Thresholds. Support threshold and confidence threshold are two important parameters in the proposed algorithm FEAST. To study how they affect the performance of FEAST, in this part, we give the sensitivity analysis of these two parameters on FEAST in terms of classification accuracy, proportion of selected features and runtime, respectively.

Classification Accuracy. Fig. 1 shows sensitivity analysis results of the support and confidence thresholds on the classification accuracies of the three classifiers with respect to our proposed algorithm FEAST.



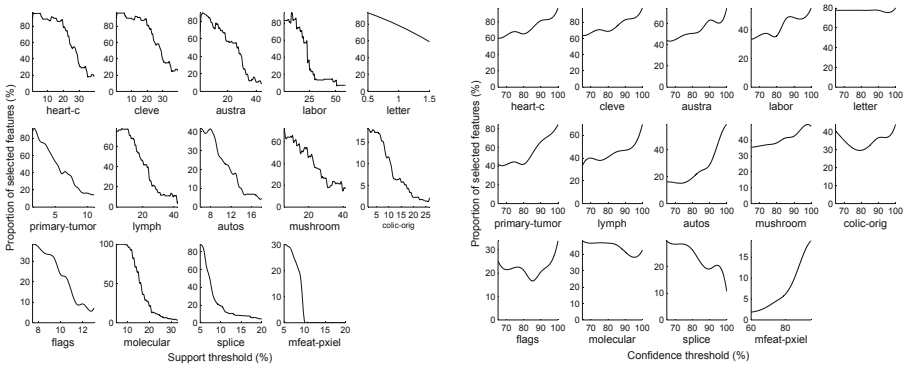
(a) Accuracies vs. Support threshold (b) Accuracies vs. Confidence threshold

Fig. 1. Classification accuracies of the three classifiers with FEAST vs. different thresholds

From Fig. 1(a) and 1(b) we observe that (i) for a given data set, the classification accuracy varying trends of the three classifiers w.r.t FEAST are very similar for either the given support thresholds or the given confidence thresholds. This reveals that the FEAST has no bias for a special classifier, i.e. the results obtained by FEAST are generally suitable. (ii) The classification accuracy varies with both the support and confidence thresholds, and the thresholds corresponding to the highest classification accuracy are different for different data sets. For example, in Fig. 1(a), the support threshold corresponding to the highest classification accuracy is about 10% for "austra", while less than 5% for "colic-orig". In Fig. 1(b), the confidence threshold corresponding to the highest classification accuracy is greater than 95% for "autos", while about 70% for "splice". This implies that both support and confidence thresholds affect the feature subset selected by FEAST, and the best thresholds are different for different data sets.

Proportion of Selected Features. Fig. 2 shows sensitivity analysis results of the support and confidence thresholds on the proportion of features selected by the proposed algorithm FEAST.

From Fig. 2(a) we observe that for all the 14 data sets, with the increment of the support threshold, the proportion of the selected features decreases. The



(a) Proportion of selected features vs. Support threshold

(b) Proportion of selected features vs. Confidence threshold

Fig. 2. Proportion of features selected by FEAST vs. different thresholds

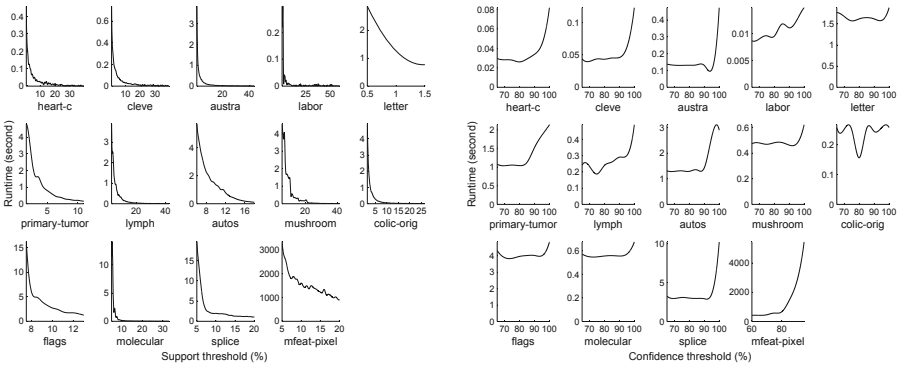
reason is that with the increment of the support threshold, the number of the frequent itemsets decreases. At the same time, FEAST chooses feature subset from itemsets that are at least frequent, thus the number of the selected features deceases, and the proportion of the selected features decreases as well. We also observe that although the proportion of the selected features deceases with the increment of the support threshold, for the different data sets, the decrement extents are varying. Therefore, we should choose different support thresholds for the different data sets.

From Fig. 2(b) we observe that with the increment of the confidence threshold, the proportion of selected features either increases or decreases. The reason is that for a given confidence threshold, there are many support thresholds with varying values. Further, for the different confidence thresholds, the varying ranges of the support thresholds are different. This means the corresponding numbers of the frequent itemsets and further the proportions of selected features are different as well. This reveals that both the support and confidence thresholds are affected by data set characteristics and we should select different thresholds for different data sets.

Runtime Fig. 3 shows the sensitivity analysis results of the support and confidence thresholds on the runtime of our proposed algorithm FEAST.

From Fig. 3(a) we observe that for all the data sets, the runtime of FEAST decreases when the support threshold increases. This is because with the increment of the support threshold, the number of the frequent itemsets is decreased. So the time spending on mining the frequent itemsets is decreased as well. At the same time, FEAST chooses the feature subset from the itemsets that are at least frequent, thus the time consumed in the feature subset identification is also deceased.

From Fig. 3(b) we observe that the runtime of FEAST can increase, decrease and fluctuate when the confidence threshold increases. The reason is that for a given confidence threshold, there are many support thresholds with varying



(a) Runtime vs. Support threshold (b) Runtime vs. Confidence threshold

Fig. 3. Runtime of FEAST vs. different thresholds

values. Further, for the different confidence thresholds, the varying ranges of the support thresholds are different. This means that the corresponding numbers of the frequent itemsets, and further the numbers of selected features are different as well. Thus, the time used to mine frequent itemsets and to identify feature subset is varying.

To summarize, the performance of the proposed algorithm FEAST is directly affected by the selection of these two input-parameters: support and confidence thresholds. However, the appropriate thresholds for different data sets would be different. That is, there are no specific support and confidence thresholds which are the best choice for all the data sets. We should pick up different thresholds for different data sets.

6 Conclusion

In this paper, we have presented a novel constraint association rule based feature selection algorithm FEAST. We have also compared FEAST with the other four representative feature selection algorithms, including two well-known algorithms CFS and FCBF, the algorithm INTERACT aiming at solving feature interaction, and an associative-rule-based algorithm FSBAR, upon both the five synthetic data sets and the 14 UCI data sets. The results on the synthetic data sets show that FEAST can identify relevant features and remove redundant ones while reserving feature interaction. The results on the real world data sets show that our proposed algorithm FEAST can reduce the number of features and outperforms all the other four feature selection algorithms in terms of the average accuracy improvement and the Win/Draw/Loss records of all the three different types of classifiers Naive Bayes, C4.5 and PART.

We have also conducted a sensitivity analysis of support and confidence thresholds to FEAST. The results show that the support and confidence thresholds play a fundamental role in the proposed algorithm. Moreover, for different data sets,

the appropriate thresholds could be different. Therefore, for further research, we plan to explore how to recommend the support and confidence thresholds for FEAST according to data set characteristics.

References

1. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007), http://archive.ics.uci.edu/ml/
2. Chen, G., Liu, H., Yu, L., Wei, Q., Zhang, X.: A new approach to classification based on association rule mining. Decision Support Systems 42(2), 674–689 (2006)
3. Dash, M., Liu, H.: Feature selection for classification. Intelligent Data Analysis 1(3), 131–156 (1997)
4. Dash, M., Liu, H.: Consistency-based search in feature selection. Artificial Intelligence 151(1-2), 155–176 (2003)
5. Fleuret, F.: Fast binary feature selection with conditional mutual information. Journal of Machine Learning Research 5, 1531–1555 (2004)
6. Frank, E., Witten, I.H.: Generating accurate rule sets without global optimization. In: Proceedings of the Fifteenth International Conference on Machine Learning, pp. 144–151. Morgan Kaufmann Publishers Inc. (1998)
7. Gheyas, I.A., Smith, L.S.: Feature subset selection in large dimensionality domains. Pattern Recognition 43(1), 5–13 (2010)
8. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
9. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: Proceedings of the Seventeenth International Conference on Machine Learning, pp. 359–366. Morgan Kaufmann Publishers Inc. (2000)
10. Han, J.: CPAR: Classification based on predictive association rules. In: Proceedings of the Third SIAM International Conference on Data Mining, vol. 3, pp. 331–335. Society for Industrial & Applied (2003)
11. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Mining and Knowledge Discovery 8(1), 53–87 (2004)
12. Jakulin, A., Bratko, I.: Testing the significance of attribute interactions. In: Proceedings of the 21st International Conference on Machine learning, pp. 409–416. ACM (2004)
13. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: Proceedings of the 11th International Conference on Machine Learning, vol. 129, pp. 121–129. Citeseer (1994)
14. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, vol. 1, pp. 338–345. Citeseer (1995)
15. Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., Verkamo, A.I.: Finding interesting rules from large sets of discovered association rules. In: Proceedings of the 3rd International Conference on Information and Knowledge Management, pp. 401–407. ACM (1994)
16. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97(1-2), 273–324 (1997)
17. Koller, D., Sahami, M.: Toward optimal feature selection. In: Proceedings of International Conference on Machine Learning, pp. 284–292. Citeseer (1996)

18. Kononenko, I.: Estimating Attributes: Analysis and Extensions of RELIEF. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994)
19. Li, W., Han, J., Pei, J.: CMAR: Accurate and efficient classification based on multiple class-association rules. In: Proceedings of IEEE International Conference on Data Mining, pp. 369–376. IEEE Computer Society (2001)
20. Liu, H., Setiono, R.: A probabilistic approach to feature selection-a filter solution. In: Proceedings of the 13rd International Conference of Machine learning. Morgan Kaufmann Pub. (1996)
21. Park, H., Kwon, H.C.: Extended relief algorithms in instance-based feature filtering. In: Proceedings of the 6th International Conference on Advanced Language Processing and Web Information Technology (ALPIT 2007), pp. 123–128. IEEE Computer Society (2007)
22. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
23. Scanlon, P., Potamianos, G., Libal, V., Chu, S.M.: Mutual information based visual feature selection for lipreading. In: Processings of the 8th International Conference on Spoken Language, pp. 857–860. Citeseer (2004)
24. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to data mining. Pearson Addison Wesley, Boston (2006)
25. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann Pub. (2005)
26. Xie, J., Wu, J., Qian, Q.: Feature selection algorithm based on association rules mining method. In: Proceedings of 8th IEEE/ACIS International Conference on Computer and Information Science, pp. 357–362. IEEE (2009)
27. Yu, L., Liu, H.: Feature selection for high-dimensional data: A fast correlation-based filter solution. In: Proceedings of 20th International Conference on Machine Leaning, vol. 20, pp. 856–863 (2003)
28. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research 5, 1205–1224 (2004)
29. Zhao, Z., Liu, H.: Searching for interacting features in subset selection. Intelligent Data Analysis 13(2), 207–228 (2009)

RadialViz: An Orientation-Free Frequent Pattern Visualizer

Carson Kai-Sang Leung* and Fan Jiang

Department of Computer Science, University of Manitoba, Canada
kleung@cs.umanitoba.ca

Abstract. Frequent pattern mining algorithms aim to find sets of frequently co-occurring items. Visual representation of the mining results is more comprehensible to users than the traditional long textual list of frequent patterns. Existing visualizers mostly show frequent patterns as graphs in a two-dimensional space with (x, y)-coordinates. Nowadays, in a collaborative environment, it is not uncommon for users to have face-to-face meetings when they show the graphs visualizing frequent patterns. In these situations, the viewing orientation of the graphs plays an important role as different orientations positively or negatively impact the graph legibility. A legible right-side-up graph to one user may become an illegible upside-down graph towards another user. In this paper, we propose a visualizer that uses a radial layout—which is orientation free—to show frequent patterns. Having such a visualizer is beneficial in the collaborative environment.

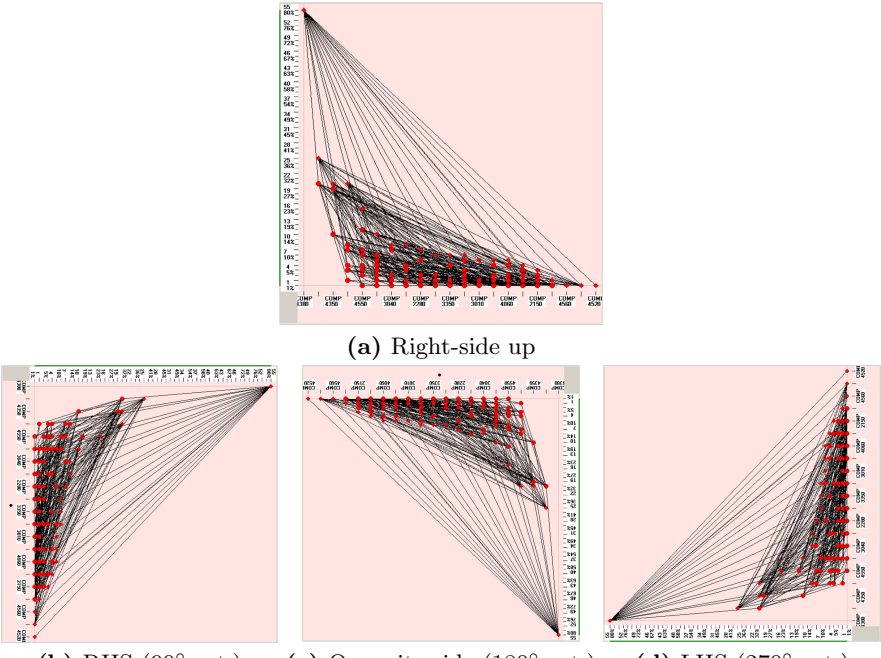
Keywords: Visual data mining, association analysis, frequent itemsets, human-machine interaction, pattern discovery.

1 Introduction

Frequent pattern mining [1] finds implicit, previously unknown, and potentially useful information in the form of sets of frequently co-occurring items or events (e.g., merchandises in a store, courses offered at a university). It plays an essential role in many knowledge discovery and data mining tasks. A common characteristic of these tasks is the identification of the frequencies of items, or sets of items, from datasets. For instance, a store manager may want to identify merchandise items that are frequently purchased together so as to place the items closer to each other (to reduce the distance required to travel by the shopper) or further apart (to encourage more purchase of items placed in between those frequently purchased ones). Similarly, a university administrator may want to know the collection of popular courses taken together by students in a semester (for lecture scheduling and exam scheduling). A book seller may want to recommend bundles of popular books to readers.

* Corresponding author.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 322–334, 2012.



(a) Right-side up

(b) RHS (90° rot.) **(c)** Opposite side (180° rot.) **(d)** LHS (270° rot.)

Fig. 1. Viewing frequent patterns with FIsViz [16] at different orientations

Over the past two decades, data mining researchers have designed and developed numerous frequent pattern mining algorithms. However, many of these algorithms have been focused on either functionality or efficiency. These algorithms usually return the mining results in textual form (e.g., a very long list of frequent patterns). Consequently, users may not easily comprehend the knowledge and useful information from the textual list. Conversely, visual representation of these patterns would be more comprehensible to users. However, not too many visualization tools have been developed to support frequent pattern mining. A common characteristic among the visualizers that were designed to support frequent pattern mining (e.g., FIsViz [16], PowerSetViewer [19]) is that they display the mined frequent patterns in a traditional two-dimensional rectangular space. For instance, FIsViz lists domain items on the x-axis, shows frequency values on the y-axis, and visualizes frequent patterns as polylines drawn on this two-dimensional rectangular space with (x, y)-coordinates. As such, the orientation of the graph displaying the patterns plays an important role in legibility of the graph. Consider a situation in which two users are facing each other and are discussing the frequent patterns shown on the graph (e.g., a marketing analyst was asked by a store manager, who sits on the opposite side of a table, to discuss the sets of merchandise items that are frequently purchased by shoppers). When showing frequent patterns in the graph as supporting evidence, it may be right-side up to the manager (e.g., as shown in Fig. 1(a)) but upside down to

the analyst (e.g., as shown in Fig. 1(c), from which important information such as frequency is not easy to read) and vice versa.

To summarize, the users who face the unfavourable orientation may have difficulty in comprehending the frequent patterns shown on the graph. To improve the situations, we propose in this paper a visualizer—called RadialViz—that uses a radial layout to visualize frequent patterns. The *key contribution* of this paper is our radial visualizer that shows the discovered frequent patterns in an orientation-free environment.

This paper is organized as follows. The next section provides background and discusses related work. We propose our visualizer in Sect. 3. Evaluation results are presented in Sect. 4. Finally, we present the conclusions in Sect. 5.

2 Background and Related Work

Development of effective visualization systems for data mining has been the subject of many studies. This line of research can be sub-classified into two general categories: (i) systems for visualizing data (e.g., VisDB [12], independence diagrams [4], Polaris [21]) and (ii) systems for visualizing the mining results (e.g., systems that visualize decision trees [3], association rules [5,11], and clusters [10,14]).

Recently, some tools and techniques have been designed to visualize patterns involving sets of items or related co-occurring entities [6,15,17,18]. For example, Wong et al. [23] designed visualization tools for visualizing topic association rules and sequential patterns appearing in documents. Their visual tools are similar to parallel coordinates, in which keywords appear on the parallel coordinate axes in the y-direction and the sequential index (temporal or others) on the x-axis.

Similarly, Yang [24,25] designed a system mainly to visualize association rules (but can also be used to visualize frequent patterns) in a two-dimensional space consisting of parallel vertical axes. In his system, all domain itemset are sorted according to their frequencies and evenly distributed along each vertical axis. A frequent pattern consisting of k items (i.e., a k-itemset) is then represented by a curve that extends from one vertical axis to another connecting k such axes. As the frequency of such a pattern is indicated by the thickness of the curve, it is not easy to compare the frequencies of patterns.

PowerSetViewer (PSV) [19] provides users with guaranteed visibility of frequent patterns in the sense that the pixel representing a frequent pattern is guaranteed to be visible by highlighting such a pixel. However, multiple frequent patterns may be represented by the same pixel, and PSV does not show the relationship between related frequent patterns (e.g., it is not easy for users to spot the prefix/extension relationship among patterns $\{a\}, \{a, b\}$ and $\{a, b, c\}$). Note that $\{a\}$ and $\{a, b\}$ are *prefixes* of $\{a, b, c\}$. Equivalently, $\{a, b, c\}$ is an *extension* of $\{a, b\}$, which is then an extension of $\{a\}$. For any k-itemset Z in a domain of m items, there are $k - 1$ non-empty prefixes (i.e., not counting the empty set and Z itself) of Z and at most $2^{m-k} - 1$ extensions of Z.

FIsViz [16] was proposed in PAKDD 2008 to visualize frequent k-itemsets as polylines connecting k nodes in a two-dimensional space with (x, y)-coordinates,

in which domain items are listed on the x-axis and frequency values are indicated by the y-axis. The x-locations of all nodes in the polyline indicates the domain items contained in a frequent pattern Z, and the y-location of the rightmost node of a polyline for Z indicates the frequency of Z. As such, prefix/extension relationships can be observed by traversing along the polylines.

Nowadays, in a collaborative environment, it is not uncommon for collaborators to have face-and-face meetings. Partially due to the emerging of tabletop displays as an effective platform for collaboration, information is shared on the tabletop surface in the meetings. As such, orientation or view perspective cannot be neglected. Unlike a single-user environment (where orientation may not be an issue), object orientation becomes critical in a multi-user environment because not all users share a common perspective of the displayed information. As information is viewed from different positions, it may be perceived differently. A recent study [2] showed that user perception (e.g., legibility or readability) of a chart decreases when the chart is not oriented right-side up. Let us use FIsViz as an example. When frequent patterns showed by FIsViz are rotated 90° or 270° clockwise (as shown in Figs. 1(b) and 1(d)) corresponding to the guests who sit on the right-hand-side (RHS) or the left-hand-side (LHS) of the host, guests may encounter difficulties in quickly reading the information. It may take much longer when the charts are put upside down (as shown in Fig. 1(c)). Hence, although FIsViz visualizes frequent patterns, it is *not* orientation free.

The aforementioned study [2] suggested that the legibility or readability can be improved by using radial charts. The sunburst technique [20,22] is a space-filling visualization that uses a *radial* layout (i.e., a ring chart, a multilevel pie, or concentric circles) [7] to offer an explicit portrayal of a hierarchical structure. Specifically, items in a hierarchy are laid out radially in sunburst. The root/top of the hierarchy is put at the center, and deeper/leaf levels are put farther away from the center (i.e., with the hierarchy moving outward from the center). Each hierarchical level forms a "block arc" or pie segment. An inner block arc (or a pie segment of an inner ring) bears a hierarchical relationship to those outer block arcs (or pie segments of an outer ring) which lie within the angular sweep of the parent arc. The arc length (and thus the central angle and area) of a block arc is usually proportional to the quantitative values associated with that arc. Fig. 2 shows how sunburst visualizes a hierarchical structure of customers. From this figure, we observe the following properties of hierarchical data represented by sunburst:

P1. All children of a node in the hierarchy are disjoint.
P2. The quantitative value associated with a parent node is higher than or equal to the *sum* of quantitative values associated with *all* its child nodes.
P3. Given P1 and P2, the quantitative value associated with a parent node is higher than or equal to the quantitative value associated with *each* of its child nodes.

Data in the hierarchical structure of customers in Fig. 2 possess the above three properties. For instance, a customer is either a member or a non-member. A member is either a gold, silver, or bronze member. Here, the quantitative value is

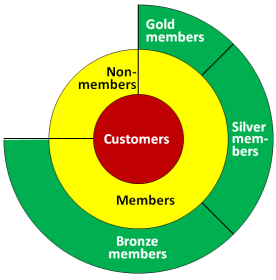


Fig. 2. An example of sunburst [20,22]

the count. The total customer count is the sum of the total numbers of members and non-members. The total membership count is the sum of the membership counts of all gold, silver, and bronze members.

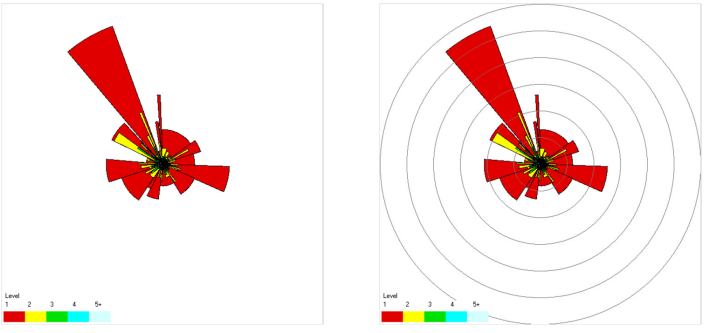
FP-Viz [13] uses a radial layout for visualizing tree-based frequent pattern mining. Specifically, FP-Viz visualizes the FP-tree [9] used in the mining process. As (i) all children of a node in an FP-tree are disjoint and (ii) the support value of a parent node is higher than or equal to the *sum* of *all* support values of its child nodes, then (iii) the support value of a parent node is higher than or equal to that of *each* of its child nodes. In other words, when FP-VIz visualizes the database transactions in the radial layout, the database transactions captured in the FP-tree satisfies Properties P1–P3 above. However, FP-Viz does *not* directly visualize frequent patterns, which need to be mined from the FP-tree. Moreover, a pattern Z may be embedded in multiple paths of an FP-tree (e.g., $\{b, e\}$ may be contained in paths representing transactions $t_i = \{a, b, c, d, e, f\}$ and $t_j = \{b, c, e, g\}$), and thus appears in different block arcs in FP-Viz. Consequently, it may not be easy to directly read the frequency of Z.

3 RadialViz: Our Proposed Visualizer

Recall from the previous section that FIsViz visualizes frequent patterns as polylines in a two-dimensional rectangular space, but FIsViz is not orientation free. In contrast, FP-Viz is orientation-free with a radial layout, but it shows the contents of an FP-tree (i.e., database transactions to be mined) instead of directly showing frequent patterns (i.e., the results mined from the FP-tree). In this section, we propose a visualizer—called ***RadialViz***—to use a radial layout (which is orientation free) to directly show frequent patterns and their relationships (e.g., prefix/extension relationships).

Visualizing the hierarchical structure of frequent patterns (and their prefix/extension relationships) in a radial layout is challenging because frquent patterns in the prefix/extension hierarchy does *not* satisfy Properties P1 and P2. We observe the following with *frequency* being the quantitative value.

1$^-$. Not all extensions of a frequent pattern Z are disjoint. In fact, extensions of Z are usually overlapping (e.g., as two extensions of $\{a, b\}$, both $\{a, b, c\}$ and $\{a, b, d\}$ are overlapping).



(a) RadialViz **(b)** RadialViz with radial grid

Fig. 3. RadialViz shows the same set of frequent patterns as in Fig. 1

2^-. The frequency of a frequent pattern Z is *not* necessarily higher than or equal to the sum of frequencies of *all* extensions of Z.

3^+. Fortunately, the frequency of a frequent pattern Z is still higher than or equal to the frequency of *each* extension of Z.

We design RadialViz for visualizing frequent patterns based on these observations.

3.1 Basic Representation of Frequent Patterns in RadialViz

Recall that both sunburst and FP-Viz divide the central angle according to quantitative values associated with the nodes in a hierarchy. This works well for the hierarchy that satisfies Properties P1 and P2 (e.g., hierarchy in an FP-tree), in which (i) child nodes of a parent node are disjoint and (ii) the quantitative values associated with parent nodes are bounded below by the sum of quantitative values associated with their child nodes (e.g., the support value of a parent node in a tree path representing a set of similar transactions is at least the sum of support values of all its child nodes in an FP-tree). The central angle for each parent node is then subdivided according to the quantitative values associated with its child nodes. However, these two properties do not hold for visualization of frequent patterns as noted in Observations 1^- and 2^-. For example, if frequency of $\{a, b\}$ is 10, then individual frequency of its extension $\{a, b, c\}$ or $\{a, b, d\}$ is at most 10. However, their sum can range from 0 to 20. If the sum were above 10, then how can we represent $\{a, b, c\}$ and $\{a, b, d\}$ radiating from the sector or block arc representing $\{a, b\}$? A naive solution is to overlap the areas for these two extensions. This works for this particular example. However, what if there are multiple extensions of $\{a, b\}$ (e.g., for a domain of 100 items, there are potentially $2^{98} - 1 \approx 3 \times 10^{29}$ extensions of $\{a, b\}$ including potentially 98 immediate extensions of $\{a, b\}$). It is unclear how to overlap these 98 extensions so that the outcome is still comprehensible to users. As such, we cannot divide the central angle according to the frequency of a frequent pattern. This leads to two questions: (i) How to represent the frequency of a pattern, which plays an

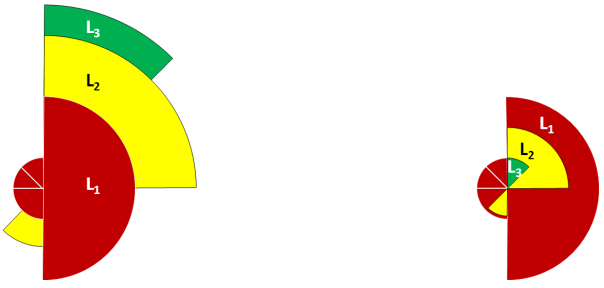
important role in frequent pattern mining? (ii) How to divide the central angle of the radial layout?

Representation of Frequency of a Frequent Pattern. To answer the first question, our *RadialViz uses radius* (instead of the central angle, sector area, or arc length) *to represent the frequency information.* By doing so, users can easily infer the frequency distribution of all frequent patterns. For example, one can easily spot from Fig. 3 the most popular course (with the high frequency) as it is indicated by the sector with the longest radius (on the upper left portion of the graph). Moreover, patterns with the same frequency have the same radius.

Representation of Cardinality of a Frequent Pattern. Representing frequency of patterns by radius may lead to the following question. In both sunburst and FP-Viz, each level of the hierarchy forms a ring or block arc. Here, when visualizing frequent patterns in RadialViz, each level of the hierarchy represents the cardinality k of k-itemsets. Given that RadialViz represents frequencies of patterns by radius, patterns of the same cardinality may not necessarily form a ring or block arc with the same radius from the center. The block arc for a k-itemset (e.g., with frequency=55) may appear much further away from the center than that for another k-itemset (e.g., with frequency=5). Since the block arcs for $(k+1)$-itemset extensions of a k-itemset Z is shown to be radiating from the block arc for Z, users can count the number of levels of block arcs to determine the cardinality of Z.

For user convenience, *RadialViz uses colour to represent the cardinality of frequent patterns.* By doing so, users can directly get the cardinality without counting multiple block arcs, each representing a cardinality level in the hierarchy. See Fig. 3, in which the colour bar at the bottom indicates the cardinality (e.g., from red indicating the minimum cardinality of 1 to light blue indicating the cardinality of 5^+ for the illustrative student database).

Representation of a Frequent Pattern. Recall that RadialViz uses colour to represent the cardinality of frequent patterns and uses radius to represent the frequency. Based on Observation 3^+, we know that the frequency of any extension of a frequent k-itemset Z is bounded above by the frequency of Z. This implies that the radii of block arcs for extensions of Z is bounded above by the radius of block arc for Z. Hence, we do not have to put the block arcs for $(k+1)$-itemset extensions of Z radiating from and *outside* the block arc for Z (as shown in Fig. 4(a)) so as to avoid having a radial graph spanning too far from the center. Instead, RadialViz stacks the block arcs or sectors for $(k+1)$-itemset extensions of Z on *top* of the block arc or sector for Z. By doing so, *RadialViz represents each frequent pattern by a sector radiating from the center.* There are several advantages of this representation of frequent patterns by RadialViz. First, the span of the radial graph is bounded above by the maximum radius of all singletons (i.e., 1-itemsets). See Fig. 4(b). Second, the prefix/extension relationship can then be represented through *containment* (i.e., sectors for the extensions of Z are contained in the sector for Z). Third, it is much easier to



(a) Radiating out level-by-level **(b)** Stacking up levels (as in RadialViz)

Fig. 4. Representation of frequent patterns

spot the change of frequency between Z and its extensions because sectors (for Z and its extensions) are all radiating from the same center.

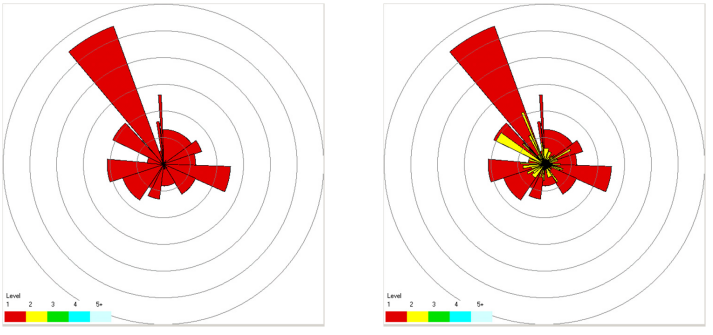
Next, let us answer the earlier question on how to divide the central angle of the radial layout. A naïve approach is to divide the central 360° angle among all singletons, and then recursively subdivide the angle associated for each k-itemset among their immediate $(k+1)$-itemset extensions. However, a potential problem associated with this naïve approach is that some sectors may be very dense (due to the large number of frequent pattern extensions) while others may be very sparse (due to the small number of frequent pattern extensions). Hence, our RadialViz uses a different approach. Instead, *RadialViz divides the central 360° angle into p sectors, and each sector represents one of the p mined frequent patterns.*

3.2 Other Features and Observations on RadialViz

In the previous section, we introduced some essential features of our RadialViz. In this section, let us present some optional features of RadialViz.

Frequency of Frequent Patterns. Recall that RadialViz uses radius to represent frequency. Sectors with long radii represent frequent patterns with high frequencies, while sectors with short radii represent less frequent patterns. In many real-life applications, users need to compare frequencies of different patterns. *RadialViz provides users with the radial gridline* so that users can easily read off the frequency of different patterns and compare among them. For example, each ring formed by the radial gridline indicates a frequency increment of 10 in Fig. 3(b). So, users can easily learn that the enrolments of the two most popular courses are 55 and 26. There are also two courses with the same enrolment of 21.

Moreover, *RadialViz also provides users details-on-demand by allowing them to hover the mouse over a sector* to get a small box showing the frequency of the corresponding frequent pattern represented by the sector. For example, the longest red sector on the upper left indicates that the corresponding course was taken by most students. When users hover the mouse over such a long red sector, the small textbox appears and explicitly shows "55" as the exact frequency for that course.



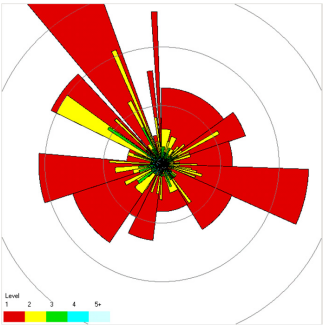
(a) Only frequent 1-itemsets **(b)** Frequent 1- and 2-itemsets

Fig. 5. Our proposed RadialViz showing frequent patterns of the first k levels/cardinalty (where k = (a) 1 and (b) 2) in Fig. 3

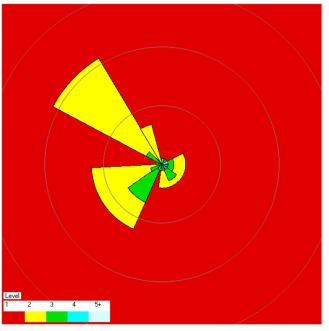
Ordering of Items in Patterns. Recall that FIsViz arranges domain items in descending frequency order. Such an ordering would be helpful if users just want to find out items with the highest or lowest frequencies. However, in many other real-life applications, it is more common for users to look up the frequency of some particular patterns of interest. In those applications, having the domain items arranged in descending frequency order means that users need to perform a linear scan for the interesting patterns. Hence, to facilitate easy lookup of frequent patterns, *RadialViz arranges items clockwise in some user-specified order* (e.g., alphabetical order). With such an arrangement, users can easily locate the patterns of interest. For instance, Figs. 3 and 5 show second- to fourth-year courses (arranged by course number clockwise from the 12 o'clock position). Users can infer some knowledge like "the most popular course is a fourth-year course". Moreover, knowing that the most popular course is COMP 4380 (with longest radius), if users want to find COMP 4350, then they only need to search in a counterclockwise direction for a sector very close to COMP 4380. Furthermore, with this item arrangement, users can still easily spot the patterns with the highest or lowest frequencies. The reason is that, as RadialViz uses radius to show frequency, patterns with the highest and lowest frequencies would have the longest and shortest radii, respectively.

Patterns of Some Specific Cardinality. Recall that RadialViz uses colour to represent the cardinality k of frequent k-itemsets. Frequent patterns of the same cardinality are represented by the same colour. Moreover, in many real-life applications, it is uncommon to find frequent patterns of certain cardinality. Hence, *RadialViz allows users to specify which cardinality levels to be displayed.* For instance, Fig. 5(a) shows only frequent patterns of cardinality 1 (i.e., 1-itemsets), whereas Fig. 5(b) shows only frequent patterns of both cardinalities 1 and 2 (i.e., 1- and 2-itemsets).

Zoom-In and Zoom-Out. When RadialViz shows all frequent patterns (as in Fig. 3), it gives users an overview about the distribution of all frequent patterns.



(a) Zooms in to the sector of interest



(b) Zooms in to drilled-in sector of interest

Fig. 6. Our proposed RadialViz (a) zooms in and (b) drills in to the sector of interest in Fig. 3

As some sectors are small, RadialViz provides users with interactive features to *zoom in* and *zoom out* so that users can obtain information of the granularity level of their interest. See Fig. 6(a) for a zoom-in view.

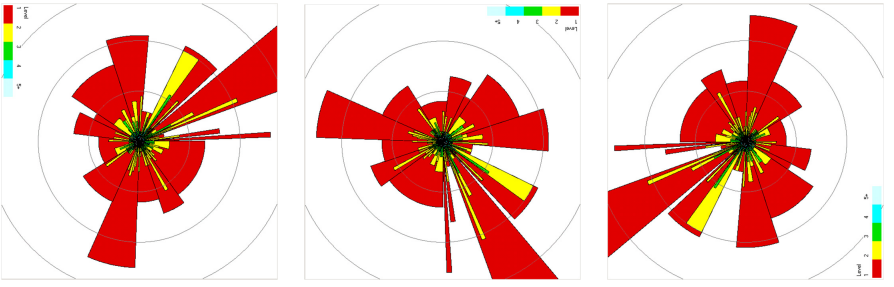
Drill-In. Moreover, RadialViz also provides users with interactive features to *drill in* some specific area of interest. The key difference between zoom-in and drill-in is that the former just magnifies the sector of interest (i.e., same layout) whereas the latter redraws the sector of interest. To get a close-up of the drilled-in image, RadialViz allows users to zoom in to the sector of interest in this drilled-in image. See Fig. 6(b) for the zoom-in view when we drilled in to details of COMP 4380. Note that combination of zoom-in and drill-in features is useful when dealing with large amounts of data.

4 Evaluation

In this section, we show our results on evaluating our proposed RadialViz. Here, we compare functionality and performance of our RadialViz with some existing systems (e.g., FIsViz [16] from PAKDD 2008). We conducted two sets of evaluation tests. In the first set, we tested functionality of our RadialViz by showing how it can be applied to various scenarios or real-life applications. In the second set, we tested performance of our RadialViz.

In terms of functionality, we considered many different real-life scenarios. We determined whether RadialViz can handle each scenarios. If so, we examined how it displays the mining results. The evaluation results show that RadialViz was effective in all these scenarios. A few samples of these scenarios are shown below:

Q1. Which course has the highest enrolment?
Q2. Which is the most frequent 2-itemset extensions of COMP 4380 and the most frequent 3-itemset extensions of COMP 4380?



(a) RHS (90° rot.) **(b)** Opposite side (180° rot.) **(c)** LHS (270° rot.)

Fig. 7. Viewing the frequent patterns in Fig. 6 with RadialViz at different orientations

Q3. How many frequent 2-itemset extensions of COMP 4380 and the most frequent 3-itemset extensions of COMP 4380 enrolled by more than 5 students?
Q4. What is the highest cardinality for the frequent extensions of COMP 4380?
Q5. Which frequent course pairs have the same frequency?

Recall from Fig. 1 the snapshots of different orientations of FIsViz. See Fig. 7 for snapshots of different orientations of RadialViz besides the right-side up view shown in Fig. 3.

With RadialViz, we easily located the course with highest enrolment (i.e., the sector with longest radius) regardless of the orientation. Although we spotted the same information from FIsViz when frequent patterns are shown right-side up, it took a bit longer time for other orientations of FIsViz.

To answer Q2 and Q3, we drilled in COMP 4380 and then zoomed in to the center (as shown in Fig. 6(b)). With the colour bar, we easily spotted the frequent 2-itemset and 3-itemset extensions (i.e., longest yellow and green sectors, respectively) of COMP 4380 from RadialViz regardless of the orientation. By hovering the mouse over the corresponding sectors, the course labels were revealed and answers were obtained (i.e., {COMP 4380, COMP 4580} with 21 students enrolled, and {COMP 4380, COMP 4550, COMP 4720} with 7 students enrolled). Similarly, we counted the number of yellow and green sectors with radius ≥ 5, and we got three course pairs and one course triplet that satisfy the enrolment condition. In contrast, for FIsViz, we needed to traverse all polylines going out from COMP 4380. As many of these polylines were bent and overlapping, it was not easy to trace and count each polyline. The situation was worsened when the graphs were not right-side up.

Similarly, for Q4, we just needed to look for the sector with colour representing the highest cardinality from the figure. In this case, it was light blue indicating four courses. For Q5, we easily spotted from Fig. 3(b) that COMP 4020 and 4350 have the same enrolment of 21. Answering these two questions in FIsViz again required traversal of those bent and overlapping polylines.

In terms of performance, we varied the size of databases. The results showed that the runtime (which includes CPU and I/Os) increased almost linearly with

the number of transactions in the database. We also varied the number of items in the domain, and the results showed that the runtime increased when the number of domain items increased. Moreover, when the user-defined frequency threshold *minsup* increased, the number of itemsets that satisfy the threshold (i.e., itemsets to be displayed) decreased, which in turn leads to a decrease in runtime. As ongoing work, we are conducting more extensive experimental evaluation.

5 Conclusions

In this paper, we proposed a frequent pattern visualization system, called *RadialViz*, which enables users to visualize the mined frequent patterns. RadialViz represents k-itemsets using a radial layout (which is orientation free) and in a hierarchical fashion (so that extensions of a pattern Z are contained within the sector representing Z). Patterns of the same cardinality have the same colour, and patterns of different cardinalities have different colours. Since RadialViz uses radius to indicate the frequencies of patterns, users can easily observe the frequency distribution of all the patterns. Patterns having similar radius have similar frequencies. With interactive features (e.g., mouse hover, zoom-in, drill-in), users can easily explore patterns of interest. Evaluation results showed the effectiveness of RadialViz. Our proposed system helps users to answer many questions for real-life applications, and thus assist them in making appropriate business intelligence (BI) decisions, especially in face-to-face tabletop collaborative environments.

Acknowledgement. This project is partially supported by NSERC (Canada) and University of Manitoba.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Advances in Knowledge Discovery and Data Mining, ch. 6. AAAI/MIT Press (1995)
2. Alallah, F., Jin, D., Irani, P.: OA-graphs: orientation agnostic graphs for improving the legibility of charts on horizontal displays. In: ITS 2010, pp. 211–220 (2010)
3. Ankerst, M., Elsen, C., Ester, M., Kriegel, H.-P.: Visual classification: an interactive approach to decision tree construction. In: ACM KDD 1999, pp. 392–396 (1999)
4. Berchtold, S., Jagadish, H.V., Ross, K.A.: Independence diagrams: a technique for visual data mining. In: KDD 1998, pp. 139–143 (1998)
5. Blanchard, J., Guillet, F., Briand, H.: Interactive visual exploration of association rules with rule-focusing methodology. KAIS 13(1), 43–75 (2007)
6. Carmichael, C.L., Hayduk, Y., Leung, C.K.-S.: Visually contrast two collections of frequent patterns. In: IEEE ICDM Workshops 2011, pp. 1128–1135 (2011)
7. Di Caro, L., Frias-Martinez, V., Frias-Martinez, E.: Analyzing the Role of Dimension Arrangement for Data Visualization in Radviz. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010, Part II. LNCS (LNAI), vol. 6119, pp. 125–132. Springer, Heidelberg (2010)

8. Grinstein, G., Plaisant, C., Laskowski, S., O'Connell, T., Scholtz, J., Whiting, M.: VAST 2008 challenge: introducing mini-challenges. In: IEEE VAST 2008, pp. 195–196 (2008)
9. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM SIGMOD 2000, pp. 1–12 (2000)
10. Hassan, M. R., Ramamohanarao, K., Karmakar, C., Hossain, M.M., Bailey, J.: A Novel Scalable Multi-class ROC for Effective Visualization and Computation. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010, Part I. LNCS (LNAI), vol. 6118, pp. 107–120. Springer, Heidelberg (2010)
11. Hofmann, H., Siebes, A.P.J.M., Wilhelm, A.F.X.: Visualizing association rules with interactive mosaic plots. In: ACM KDD 2000, pp. 227–235 (2000)
12. Keim, D.A., Kriegel, H.-P.: Visualization techniques for mining large databases: a comparison. IEEE TKDE 8(6), 923–938 (1996)
13. Keim, D.A., Schneidewind, J., Sips, M.: FP-Viz: visual frequent pattern mining. IEEE InfoVis 2005 Poster (2005)
14. Koren, Y., Harel, D.: A two-way visualization method for clustered data. In: ACM KDD 2003, pp. 589–594 (2003)
15. Leung, C.K.-S., Carmichael, C.L.: FpVAT: a visual analytic tool for supporting frequent pattern mining. ACM SIGKDD Explorations 11(2), 39–48 (2009)
16. Leung, C.K.-S., Irani, P.P., Carmichael, C.L.: FIsViz: a frequent itemset visualizer. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 644–652. Springer, Heidelberg (2008)
17. Leung, C.K.-S., Irani, P.P., Carmichael, C.L.: WiFIsViz: effective visualization of frequent itemsets. In: IEEE ICDM 2008, pp. 875–880 (2008)
18. Leung, C.K.-S., Jiang, F., Irani, P.P.: FpMapViz: a space-filling visualization for frequent patterns. In: IEEE ICDM Workshops 2011, pp. 804–811 (2011)
19. Munzner, T., Kong, Q., Ng, R.T., Lee, J., Klawe, J., Radulovic, D., Leung, C.K.: Visual mining of power sets with large alphabets. Technical report TR-2005-25, UBC, Canada (2005)
20. Stasko, J., Zhang, E.: Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In: IEEE InfoVis 2000, pp. 57–65 (2000)
21. Stolte, C., Tang, D., Hanrahan, P.: Query, analysis, and visualization of hierarchically structured data using Polaris. In: ACM KDD 2002, pp. 112–122 (2002)
22. Tobiasz, M., Isenberg, P., Carpendale, S.: Lark: coordinating co-located collaboration with information visualization. IEEE TVCG 15(6), 1065–1072 (2009)
23. Wong, P.C., Cowley, W., Foote, H., Jurrus, E., Thomas, J.: Visualizing sequential patterns for text mining. In: IEEE InfoVis 2000, pp. 105–111 (2000)
24. Yang, L.: Visualizing Frequent Itemsets, Association Rules, and Sequential Patterns in Parallel Coordinates. In: Kumar, V., Gavrilova, M.L., Tan, C.J.K., L'Ecuyer, P. (eds.) ICCSA 2003, Part I. LNCS, vol. 2667, pp. 21–30. Springer, Heidelberg (2003)
25. Yang, L.: Pruning and visualizing generalized association rules in parallel coordinates. IEEE TKDE 17(1), 60–70 (2005)

Feature Weighting by RELIEF Based on Local Hyperplane Approximation

Hongmin Cai[1,⋆] and Michael Ng[2]

[1] South China University of Technology, Guangdong, P.R. China
caihongm@sysu.edu.cn
[2] Department of Mathematics, Hong Kong Baptist University, Hong Kong

Abstract. In this paper, we propose a new feature weighting algorithm through the classical RELIEF framework. The key idea is to estimate the feature weights through local approximation rather than global measurement, as used in previous methods. The weights obtained by our method are more robust to degradation of noisy features, even when the number of dimensions is huge. To demonstrate the performance of our method, we conduct experiments on classification by combining hyperplane KNN model (HKNN) and the proposed feature weight scheme. Empirical study on both synthetic and real-world data sets demonstrate the superior performance of the feature selection for supervised learning, and the effectiveness of our algorithm.

Keywords: Feature weighting, local hyperplane, RELIEF, Classification, KNN.

1 Introduction

Feature weighting plays an important step in the preprocessing of data, especially in data classification. In general, the feature weights are obtained by assigning a continuous relevance value to each feature via a learning algorithm by stressing on the context or domain knowledge. The feature weighting procedure is particularly useful for instance based learning models, which usually construct the distance metric by using all features. Moreover, feature weighting can reduce the risk of over-fitting by removing noisy features, thereby improve the predictive accuracy. Existing feature selection methods broadly falls into two categories, wrapper and filter methods. Wrapper methods use the predictive accuracy of predetermined classification algorithms (called base classifier), such as SVMs, as the criteria to determine the goodness of a subset of features [9,15]. Filter methods select features based on discriminant criteria that relies on the characteristics of data, independent of any classification algorithm [7,14,17]. The common discriminant criteria includes entropy measurement [18], Chi-squared

⋆ This work was supported in part by NSFC under award number 60902076, NSF of Guangdong Province under award number 9451027501002551, and China Fundamental Research Funds for the Central Universities under award number 11*lgpy*33.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 335–346, 2012.

measurement [21], Fisher ratio measurement [10], mutual information measurement [20,4,3], and RELIEF-based measurement [19,27,28].

Due to the emerging needs in biomedical and bioinformatics areas, researchers are particularly interested in algorithms which can process of data with feature being of large (or huge) dimensions, such as, microarray scanning in cancer research. Therefore, filter methods are widely used due to its efficiency in computation. Among the existing filter methods in feature weighting, the RELIEF algorithm [19] is considered as one of the most successful ones due to its simplicity and effectiveness. The main idea behind RELIEF is to iteratively update feature weights by a distance margin to estimate the difference between neighboring patterns. It has been further generalized to average multiple, instead of just one, nearest neighbors when computing the sample margins, and was named as RELIEF-F [19]. The authors have shown that RELIEF-F can achieve significant improvement on performance of the original RELIEF [19]. Sun systematically proved that RELIEF is indeed an online algorithm for a convex optimization problem [27]. Through maximizing an averaged margin of nearest patterns in feature scaled space, RELIEF could estimate the feature weight in a straightforward and efficient manner. Based on the theoretical framework, one can impose outlier removal scheme called I-RELIEF since the margin averaging is sensitive to large variations [27]. To accomplish sparse feature weighting, the author introduced the l_1 penalty into optimization of I-RELIEF [28].

In this paper, we present a new feature weighting algorithm to extend classical RELIEF model. The main contribution of the proposed algorithm is that the feature weights are estimated from local patterns other than global ones, as used in exiting methods [19,27,28]. Therefore, the proposed feature weighting scheme is particularly useful when combined with local pattern based classifiers, such as HKNN [30], *ADAMENN* [8] and discriminant adaptive nearest neighbor (DANN) [16]. Besides, local patterns are more robust to the noises and outliers. It is promising to be used in applications where data are severely contaminated by noises or rich of redundance.

This paper is organized as follows. Section 2 introduces the background of the classical RELIEF method and its variations, including F-RELIEF and I-RELIEF. The main result is reported in this section. Section 3 demonstrates the performance of the proposed model. Extensive experiments have been conducted to compare with the classical methods on benchmark data sets. Conclusion is presented in Section 4.

2 The Proposed Method

2.1 RELIEF

The RELIEF algorithm has been successfully applied in feature weighing due to its simplicity and effectiveness [19,28]. The main idea of RELIEF is to iteratively adjust feature weights according to their ability to discriminate among neighboring patterns. Mathematically, suppose that $\boldsymbol{x}$ is a randomly selected sample of a binary class data. One can estimates its two nearest neighbors, wherein one

is from its same class (called *the nearest hit* or NH) and the other is from a different class (called *the nearest miss* or NM). Then the weight w_i for the i-th feature is updated by a heuristic estimation:

$$w_i = w_i + |x^{(i)} - NM^{(i)}| - |x^{(i)} - NH^{(i)}| \tag{1}$$

Since there is no exhaustive or iterative search evolved in RELIEF updating, this scheme is very efficient for the processing of data with huge dimensions, thus it is particularly promising for large-scale problems such as analysis of microarray data [24,28,7]. The authors have generalized the RELIEF model by averaging k, instead of just one in Eq. (1), nearest neighbors when computing the sample margins and was named as RELIEF-F model [19]. Experimental results have shown that RELIEF-F achieves superior performance over the original RELIEF. Its success is due to the robustness of margin estimation on multiple samples. However, the optimal number of nearest neighbors needs to be estimated empirically. Besides, RELIEF-F is also sensitive to noise degradation and the outliers. An benchmark achievement has been reported in [27], in which the author firstly proved that RELIEF is a convex optimization problem with a margin-based objective function,

$$\begin{aligned} &\max_{\boldsymbol{w}} \sum_{n=1}^{n} \rho_n(\boldsymbol{w}) \\ &\qquad := \textstyle\sum_{n=1}^{N}(\sum_{i=1}^{I} \omega_i |\boldsymbol{x}_n^{(i)} - NM^{(i)}(\boldsymbol{x}_n)| - \sum_{i=1}^{I} \omega_i |\boldsymbol{x}_n^{(i)} - NH^{(i)}(\boldsymbol{x}_n)|) \\ &\boldsymbol{s.t.} \quad \|\boldsymbol{w}\|_2^2 = 1, \qquad\qquad \boldsymbol{w} \geq 0 \end{aligned} \tag{2}$$

where $\rho_n = d(\boldsymbol{x}_n - NM(\boldsymbol{x}_n)) - d(\boldsymbol{x}_n - NH(\boldsymbol{x}_n))$ is defined as margin of a sample $\boldsymbol{x}_n$ for distance function $d(\boldsymbol{x}) = \sum_i |x_i|$. $NM(\boldsymbol{x}_n)$ and $NH(\boldsymbol{x}_n)$ are the nearest miss and hit for a sample $\boldsymbol{x}_n$, respectively.

To tackle the drawbacks of RELIEF, such as outlier detection and inaccurate updating, Sun reformulated the above problem as maximization of expected margin through scaling of features [27,28]:

$$\begin{aligned} \mathbf{E}[\rho(\boldsymbol{w})] &= \boldsymbol{w}^T(\mathop{\mathbf{E}}_{i \in NM}[|\boldsymbol{x}_n - \boldsymbol{x}_i|] - \mathop{\mathbf{E}}_{i \in NH}|[\boldsymbol{x}_n - \boldsymbol{x}_i|)] \\ &= \boldsymbol{w}^T \sum_{i \in NM} P(\boldsymbol{x}_i = NM(\boldsymbol{x}_n)|\boldsymbol{w})|\boldsymbol{x}_n - \boldsymbol{x}_i| - \sum_{i \in H_n} P(\boldsymbol{x}_i = NH(\boldsymbol{x}_n)|\boldsymbol{w})|\boldsymbol{x}_n - \boldsymbol{x}_i| \\ &= \boldsymbol{w}^T \boldsymbol{z}_n \end{aligned} \tag{3}$$

where $NM = \{i : 1 \leq i \leq N, y_i \neq y_n\}, NH = \{i : 1 \leq i \leq N, y_i = y_n, i = n\}$ are the sets of the nearest miss and the nearest hit, respectively. $P(\boldsymbol{x} = NM(\boldsymbol{x}_n)|W)$ (or $P(\boldsymbol{x} = NH(\boldsymbol{x}_n)|W)$) are the probabilities of the sample $\boldsymbol{x}$ being in the set of $NM(\boldsymbol{x}_n)$ (or $NH(\boldsymbol{x}_n)$) in the feature space scaled by weights $\boldsymbol{w}$. Though the probability distributions are unknown in prior, they can be estimated via kernel density estimation [6]. Empirical study has shown that the I-RELIEF achieves significant improvements over the traditional models. Task of classification on feature scaled dataset achieves higher accuracy than standard techniques such as SVM [12,15,9,26] and NN model [25]. Task of feature weighting is also robust

to noisy features. In applications with a huge dimension of features, economic feature weights are appreciated not only because of computational consideration, but also most features being irrelevant [14,17]. To obtain sparse and economic feature weighting, the author introduced the l_1 penalty into the optimization of I-RELIEF [28].

However, since the expectation in Eq. (3) is carried out on the set of nearest miss or hit, which consisted of the nearest neighbors of all observed samples, the feature weight estimation may be less inaccurate if the samples contain many outliers, or most of the features are being irrelevant. In both cases, the distance between the tested one and its nearest neighbors are in large value. It follows that large bias will be introduced in margin estimation via averaging operation. Although one can reduce the influence of the abnormal samples by introducing kernel distribution estimation [27,28], it will introduce additional free parameter estimation. Moreover, probability estimation via kernel approximation is sensitive to the sample size [6,13]. Therefore, it limits the empirical applications such as in analysis of microarray data, in which the data is notoriously known for that the dimension of sample observation is far less than that of the sample feature [11]. In this paper, we propose to use a local hyperplane to approximate the set of the nearest hit and miss and then estimate the feature weight through maximization of an expected margin defined by the hyperplane. The contribution of this approximation is that the hyperplane is more robust for noisy features degradation than averaging over all neighbors [19,27,28].

2.2 Approximation by Local Hyperplane

Given a sample $\boldsymbol{x}$, it can be represented by a local hyperplane of class c by:

$$LH_c(x) = \{\boldsymbol{s} \mid \boldsymbol{s} = \boldsymbol{H}\boldsymbol{\alpha}\}, \tag{4}$$

where $\boldsymbol{H}$ is a $I \times n$ matrix composed by n NNs of the sample $\boldsymbol{x}$: $\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n\}$, with $\boldsymbol{h}_i$ being the i-th nearest neighbor (called *prototype*) of class c. The parameter of $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^T$ is the weights of the prototypes $\{\boldsymbol{h}_i,\ i = 1, 2, \ldots, n\}$. It can be viewed as spanning coefficients of the subspace $LH_c(\boldsymbol{x})$. Therefore, the hyperplane can be represented as: $\{\cdot \mid \boldsymbol{H}\boldsymbol{\alpha} = \alpha_1\boldsymbol{h}_1 + \alpha_2\boldsymbol{h}_2 + \ldots + \alpha_n\boldsymbol{h}_n\}$. The value of $\boldsymbol{\alpha}$ is solved by minimizing the distance between the sample $\boldsymbol{x}$ and its local hyperplane of $LH_c(\boldsymbol{x})$ within feature scaled space:

$$J_c(\boldsymbol{\alpha}) = arg\min \frac{1}{2}\sum_{i=1}^{I} \omega(i)(x_i - s_i)^2 = \frac{1}{2}(\boldsymbol{x} - \boldsymbol{H}\boldsymbol{\alpha})^T \boldsymbol{W}(\boldsymbol{x} - \boldsymbol{H}\boldsymbol{\alpha})$$

Subject to :

$$\sum_{i=1}^{k} \alpha_i = 1 \ , \quad \boldsymbol{\alpha} \geq 0 \tag{5}$$

where $\boldsymbol{s} = (s_1, s_2, \ldots, s_I) = \boldsymbol{H}\boldsymbol{\alpha} \in LH_c(\boldsymbol{x})$. $\boldsymbol{W}$ is a diagonal matrix with diagonal elements w_i being the weight of the i-th feature.

We are proposing to use the hyper plane to represent the set of nearest miss $NM(\boldsymbol{x})$ and nearest hit $NH(\boldsymbol{x})$ for the given sample $\boldsymbol{x}$. The beneficiary of the representation is to characterize the local sample patterns robustly. Then the distance between the sample to its NH (or NM) set can be estimated from its local hyperplane other than averaging across over all samples within the set. Therefore, we redefine the margin for a sample $\boldsymbol{x}$ as $\rho_n = d(\boldsymbol{x}_n - LH_{NM}(\boldsymbol{x}_n)) - d(\boldsymbol{x}_n - LH_{NH}(\boldsymbol{x}_n))$. The feature weights are now estimated through maximization of total margins:

$$\begin{aligned}\max_{\boldsymbol{w}} \mathbf{E}[\rho(\boldsymbol{w})] &= \frac{1}{N}\max_{\boldsymbol{w}} \sum_{n=1}^{N}(\sum_{i=1}^{I}\omega_i|\boldsymbol{x}_n^{(i)} - LH_{NM}^{(i)}(\boldsymbol{x}_n)| - \sum_{i=1}^{I}\omega_i|\boldsymbol{x}_n^{(i)} - LH_{NH}^{(i)}(\boldsymbol{x}_n)|)\\ &= \boldsymbol{w}^T\frac{1}{N}\max_{\boldsymbol{w}} \sum_{n=1}^{N}(\sum_{i=1}^{I}|\boldsymbol{x}_n^{(i)} - \boldsymbol{\alpha}\boldsymbol{H}_{NM}^{(i)}(\boldsymbol{x}_n)| - \sum_{i=1}^{I}|\boldsymbol{x}_n^{(i)} - \boldsymbol{\beta}\boldsymbol{H}_{NH}^{(i)}(\boldsymbol{x}_n)|)\\ &= \boldsymbol{w}^T\boldsymbol{z}_n \end{aligned} \tag{6}$$

where $\boldsymbol{H}_{NM}(\boldsymbol{x}_n)$ and $\boldsymbol{H}_{NH}(\boldsymbol{x}_n)$ are the nearest neighbors for the set of the nearest miss and hit of the sample $\boldsymbol{x}_n$. $\boldsymbol{\alpha}_n$ and $\boldsymbol{\beta}_n$ are the coefficients for spanning of hyperplane $LH_{NM}^{(n)}$ and $LH_{NH}^{(n)}$. $\boldsymbol{w}$ is a vector with its i-th element $\boldsymbol{w}(i)$ being the weight of the i-th feature, for $i = 1, 2, \ldots, I$. To solve the minimization problem of Eq. (6), one should estimate the parameters of $\boldsymbol{\alpha}_n$, $\boldsymbol{\beta}_n$, which are dependent on the nearest neighborhoods. The main problem of the estimation, however, is that the nearest neighbors of a given sample are unknown before learning. In the presence of many thousands of irrelevant features, the nearest neighbors defined in the original space can be completely different from those in the induced space. Therefore, the nearest neighbors defined in the original feature space may not be true in the weighted feature space. To solve the difficulties, we have designed an iterative algorithm, similar to the EM algorithm and I-RELIEF [27], to achieve the goal.

Step 1: In t-th iteration, for a given sample $\boldsymbol{x}$, we estimate the parameter of $\boldsymbol{\alpha}$ by constructing the local hyperplane of the nearest hit set within induced feature space. It is trivial to show that the minimization of Eq. (5) is equivalent to solving the following quadratic programming:

$$\begin{aligned}&\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T\bar{\boldsymbol{H}}\boldsymbol{\alpha} + \boldsymbol{f}^T\boldsymbol{\alpha}\\ &s.t. \quad \mathbf{1}^T\boldsymbol{\alpha} = 1,\ \boldsymbol{\alpha} \geq 0\end{aligned} \tag{7}$$

where $\bar{\boldsymbol{H}} = \boldsymbol{H}^T\boldsymbol{W}^{(i)}\boldsymbol{H}$, $\boldsymbol{f} = -\boldsymbol{x}^T\boldsymbol{W}^{(i)}\boldsymbol{H}$, and $\mathbf{1}$ is an unitary vector whose elements are all being 1. The matrix of $W^{(i)}$ is the t-th feature weight matrix, satisfying $W^{(i)}\mathbf{1} = \boldsymbol{w}$. The parameter of $\boldsymbol{\beta}$ for nearest miss hyperplane is obtained similarly. Minimization of Eq. (5) is a constrained quadratic program problem and standard techniques can be used to obtain its solution. In particular, since

the matrix of $\bar{\boldsymbol{H}}$ is symmetric and non-negative, the minimization could be solved efficiently through standard techniques, such as active set [23].

Step 2: Estimation of the total margin with respect to $\boldsymbol{w}^{(i)}$.

$$\rho(\boldsymbol{w}^{(i)}) = \frac{1}{N}\sum_{n=1}^{N}(\sum_{i=1}^{I}\omega_i|\boldsymbol{x}_n^{(i)} - \boldsymbol{\alpha}\boldsymbol{H}_{NM}^{(i)}(\boldsymbol{x}_n)| - \sum_{i=1}^{I}\omega_i|\boldsymbol{x}_n^{(i)} - \boldsymbol{\beta}\boldsymbol{H}_{NH}^{(i)}(\boldsymbol{x}_n)|) \quad (8)$$

Step 3: Estimation of the weight $\boldsymbol{W}$ in $(i+1)$-th iteration.

$$\begin{aligned}\boldsymbol{w} &= arg\max_{\boldsymbol{w}} \rho(\boldsymbol{w}^{(i)}) \\ &= \frac{1}{N}\sum_{n=1}^{N}(\sum_{i=1}^{I}\omega_i|\boldsymbol{x}_n^{(i)} - \boldsymbol{\alpha}\boldsymbol{H}_{NM}^{(i)}(\boldsymbol{x}_n)| - \sum_{i=1}^{I}\omega_i|\boldsymbol{x}_n^{(i)} - \boldsymbol{\beta}\boldsymbol{H}_{NH}^{(i)}(\boldsymbol{x}_n)|) \quad (9)\end{aligned}$$

The above steps iterate alternatively until their convergence. The last two steps are similar to the one used in I-RELIEF [27], and we name our scheme as **LH-RELIEF** since it requires a local hyperplane approximation.

The pseudo-code for the LH-RLIEF is summarized in Alg. (2.1)

Algorithm 2.1: LH-RELIEF ALGORITHM(V, W, λ)

comment: Variables Initialization: $\boldsymbol{w}$=$\frac{1}{I}$, stopping criteria ϵ and number of iterations T

for $t \leftarrow 1$ **to** T
while $\|\boldsymbol{w}^{(t+1)} - \boldsymbol{w}^{(t)}\| > \epsilon$

do
1. Estimate the coefficients for hyperplane of nearest miss and hit $\boldsymbol{\alpha}, \boldsymbol{\beta}$
2. Calculate the margin by Eq .(8)
3. Update the weights by Eq. (9)

return $(\boldsymbol{w})$

3 Experimental Results

We shall demonstrate the performance of the proposed scheme through classification evaluation on both synthetic and empirical problems. In particular, we are interested in its: 1) performance of classification compared with other feature weighting scheme; 2) robustness when processing the samples with irrelevant features of large dimension.

3.1 Selection of Classifier

In our experiments, we selected the hierarchical k-nearest neighbor (HKNN) algorithm to conduct the comparison on feature weighting [30]. HKNN could be viewed as a localized approximation of K-nearest neighbor model. In this model, each class is modeled as a smooth and low-dimensional manifold embedded in the high-dimensional data space by assuming that the manifolds are locally linear.

There are two steps involved in classification by HKNN. In the first step, for each tested sample, it constructs local hyperplanes for each class. The label of the tested sample is assigned to the class whose local hyperplane to the tested sample is minimized. Empirical study has shown that the HKNN produced a comparable or even better performance of classification than standard techniques, including KNN and SVM [30,8,29]. One may note that the HKNN model shares the similar idea with our approach in that the sample information is inferred from local structure, which is the main reason for us to choose this particular classifier.

Since the HKNN model does not consider the influence of feature weights, the test data will be firstly scaled into feature space before the classification is carried out. The hyper-parameters used in training phase are estimated through ten-fold cross validation.

3.2 Fermat's Spiral Problem

In the first example, we shall test the performance of the proposed method on the well-known Fermat's Spiral problem. The test dataset consists of two classes with 200 samples for each class. The labels of the Spiral are completely determined by its first two features. The shape of the Fermat's Spiral distribution is shown in Fig. 1(a). Heuristically, the label of a sample will be inferred easily from its local neighbors. Classification based on local information will give more accurate assignment than global measurement based prediction (or classification) does since the later one is sensitive to noise degradation. To tackle this drawback, Sun proposed to lower the influence of the samples nearby through modeling of their probability distribution via kernel techniques [27]. This strategy is straightforward and successful. However, if the dominant (informative) features are buried by the irrelevant (less informative) ones, estimation of the probability via distance will be less accurate since the irrelevant feature may introduce a large variation to distance, for instance, the irrelevant features are being in a huge dimension. In order to show this, irrelevant features following standard norm distribution are added to the Spiral for classification testing. The dimensions of irrelevant features are ranging from $\{0, 300, 600, 900, 1200, 1500, 1800, 2100, 2400, 2700, 3000\}$. Two feature weighting scheme, I-RELIEF and LH-RELIEF were firstly applied to quantify the importance of feature. Then the classification was performed on dataset scaled by the feature weights. For each experiment, ten folds cross validation scheme is used to compute the accuracy of classification. To eliminate the statistical variations, we have conducted ten times experiments independently on each dataset and averaged classification error is recorded and, shown in Fig. 1(b). We observe that, the performance of the two methods are very similar when the dimension of the irrelevant features is small. However, if the dimension of irrelevant features tends to be large, the performance of I-RELIEF is severely degraded by the noises. In comparison, the performance of LH-RELIEF is very stable and produces superior outcomes.

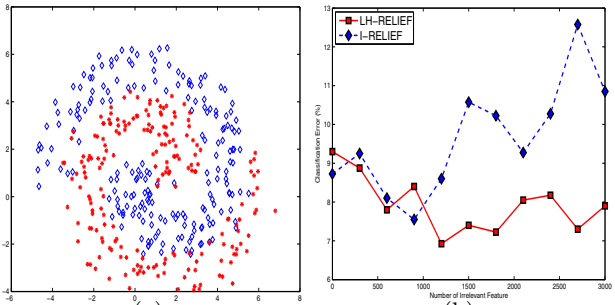


Fig. 1. Experiment on Fermat's Spiral. (a) Distribution of binary Fermat's Spiral problem. Each class has 200 samples and is labeled by different colors; (b) Irrelevant features with verified dimensions are added to test the robustness of the feature weighting schemes. The LH-RELIEF outperforms I-RELIEF with respect to classification error. With the increase of dimension of the irrelevant features, the performance of I-RELIEF is degraded while LH-RELIEF keeps stable.

3.3 UCI Data Sets

In the second experiment, we tested the proposed technique on ten medium sized datasets. The tested benchmark data sets were downloaded from the UCI Machine Learning Repository [1], and they have been widely tested by various classification benchmark models. The characteristics of the datasets are summarized in Table 1. We compare our algorithm with four other algorithms, including Iterative Search Margin Based Algorithm (Simba) [2], sparse Bayesian multinomial logistic regression (SBMLR) [5] and I-RELIEF [27]. Simba is a local learning based algorithm similar to RELIEF. SBMLR is a special kind of sparse multinomial logistic regression models with Bayesian regularization. Multinomial logistic regression algorithm has been successfully used in text processing [31] and microarray classification [22]. The beneficiary of adding regularization parameter into sparse multinomial logistic regression via a Laplace prior is that an analytical solution could be obtained. Besides, its performance is similar to using cross-validation based model selection, thus greatly reducing computational expense.

For each dataset, the optimal parameters were estimated by ten-fold cross validation. The obtained feature weights under optimal parameters were used to scale the raw datasets. Twenty times experiments on each dataset were performed independently and classification errors were averaged to evaluate the performance of the feature weighting scheme. We will use the classification error to quantify the discrimination power of weighting scheme. Furthermore, statistical testing is also useful to fully comprise the performance of feature weights [27]. We selected the Students paired two-tailed t-test to achieve the goal. The p-value of the t-test represents the probability that two sets of compared results come from distributions with an equal mean. In this experiment, a p-value of 0.05 is considered statistically significant.

The results are summarized in Table 2. We observe that that LH-RELIEF and I-RELIEF are statistically different from the tested ten datasets. The performance of classification after LH-RELIEF is better than after I-RELIEF in 9 of 10 experiments. Among the four feature weighting schemes, LH-RELIEF outperforms others in 5 of 10 datasets, while almost is suboptimal in other five dataset.

Table 1. Summary of tested datasets and their characteristics

Data set	#Instances	#Classes	#Feature
Bupa	345	2	6
Teach	151	3	5
Sonar	208	2	60
Cancer	198	6	32
Prokaryotic	997	3	20
Eukaryotic	2427	4	20
Haberman	306	2	3
Page block	5473	5	10
Pima	768	2	8
Spambase	4601	2	57

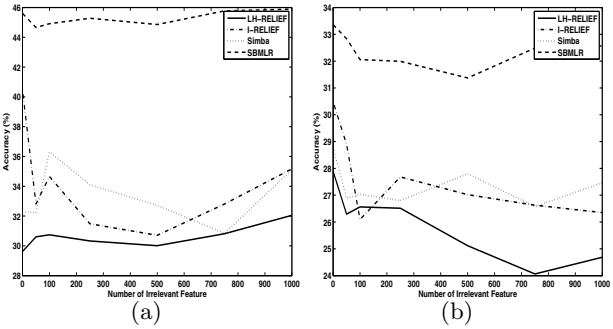


Fig. 2. Experiment on benchmark dataset of Bupa and Pima by adding irrelevant features in verified dimension, extending from 0 to 1000. (a) Bupa; (b) Pima.

In the last experiment, we are willing to test the performance of the algorithm on data in huge dimensions. More specifically, we are interested in the robustness of the algorithm on feature weighting with respect to the dimension of the irrelevant features. We selected two test datasets: Bupa and Pima. For each dataset, irrelevant features are added to the raw dataset. The added irrelevant features are independently sampled from zero-mean and unit-variance Gaussian distribution. Their dimensions are ranged from 0 to 1000. Including useless features is

Table 2. Classification accuracies (%) on 10 real data sets. The LH-RELIEF shows to be statistically different from the I-RELIEF in 9 among 10 datasets. The P-value for each dataset is shown in parenthesis. Overall, the better results are subscripted by star under different feature weighting scheme. The LH-Relief outperforms the standard ones in most cases when the two methods show a statistically difference.

Dataset	LH-RELIEF	I-RELIEF (P-value)	SBMLR	Simba
Bupa	69.7*	66.7 (0.00)	56.2	66.8
Teach	64.4*	46.3 (0.00)	34.4	62.3
Sonar	86.7*	84.3 (0.00)	82.7	85.7
Cancer	76.2	76.0 (0.48)	76.9*	76.4
Prokaryotic	90.5*	89.8 (0.00)	90.4	89.3
Eukaryotic	82.8	81.2 (0.00)	83.5*	81.3
Haberman	69.3	72.3 (0.00)	69.9	68.7
Page Block	94.5	94.1 (0.00)	95.7*	89.8
Pima	74.0	70.3 (0.00)	68.9	74.5*
Spambase	84.8*	78.0 (0.00)	79.3	39.4

less appreciated in applications where the acquisition of data is quite expensive. For example, it may complicate the pathway research if irrelevant genes are included in microarray data analysis [27]. We would welcome such complication in order to show the robustness of the algorithm.

The hyper-parameters, such as the kernel size σ in I-RELIEF and the number of nearest neighbors k in LH-RELIEF are estimated through ten-fold cross validation. To eliminate statistical variations, each algorithm is run for twenty times on each noisy dataset. In each run, a dataset is randomly partitioned into training and testing. The averaged testing errors serve as the criterion to quantify the performance of the algorithm, and the results are drawn in Fig.2. For Bupa, the classification error of the classifier after LH-RELIEF is smaller than that after I-RELIEF in all dimensions, Fig. 2(a). This observation is coincided with the results in Table. 2, implying that the feature weights estimated by LH-RELIEF are more accurate and robust to the noises. For Pima, the performance of the two scheme is almost comparable when the dimension of the the irrelevant features is small, Fig. 2(b). However, the testing error after LH-RELIEF dramatically decreased with respect to the dimension of the irreverent features. In comparison, the classification error after I-RELIEF tends to be greater. The experiment further demonstrates that the proposed feature weighting scheme is more immune to the noisy features by showing surprising high degree of robustness.

4 Discussion

In this paper, we proposed a new feature weight scheme to tackle the common drawbacks of the RELIEF family. The nearest miss and hit subset are

approximated by constructing a local hyperplane. Then the updating of feature weights is achieved by measuring the margin between the sample and its hyperplane under general RELIEF framework. The main contribution of the new variation is that the margin is more robust to the noises and the outliers than earlier works do. Therefore, the feature weights can characterize the local structure more accurately. Experimental results on both synthetic and real-world datasets validate our findings. The proposed weighting scheme performs superior on most test data with respect to classification error. We also observed that the algorithm was convergent in most cases, though theoretical justification is needed.

References

1. Asuncion, A., Newman, D.: UCI machine learning repository (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
2. Bachrach, G.R., Navot, A., Tishby, N.: Margin Based Feature Selection - Theory and Algorithms. In: Proc. 21st International Conference on Machine Learning (ICML), pp. 43–50 (2004)
3. Brown, G.: An Information Theoretic Perspective on Multiple Classifier Systems. In: Benediktsson, J.A., Kittler, J., Roli, F. (eds.) MCS 2009. LNCS, vol. 5519, pp. 344–353. Springer, Heidelberg (2009)
4. Brown, G.: Some Thoughts at the Interface of Ensemble Methods and Feature Selection. In: El Gayar, N., Kittler, J., Roli, F. (eds.) MCS 2010. LNCS, vol. 5997, pp. 314–314. Springer, Heidelberg (2010)
5. Cawley, G.C., Talbot, N.L.C., Girolami, M.: Sparse Multinomial Logistic Regression via Bayesian L1 Regularisation. Advances in Neural Information Processing Systems 19 (2007)
6. Christopher, A., Andrew, M., Stefan, S.: Locally weighted learning. Artificial Intelligence Review 11, 11–73 (1997)
7. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. Journal of Bioinformatics and Computational Biology 3(2), 185–205 (2005)
8. Domeniconi, C., Peng, J., Gunopulos, D.: Locally adaptive metric nearest-neighbor classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(9), 1281–1285 (2002)
9. Duan, K.B.B., Rajapakse, J.C., Wang, H., Azuaje, F.: Multiple SVM-RFE for gene selection in cancer classification with expression data. IEEE Transactions on Nanobioscience 4(3), 228–234 (2005)
10. Duda, R., Hart, P., Stork, D.: Pattern Classification. Wiley (2001)
11. Fraley, C., Raftery, A.E.: Model-based clustering, discriminant analysis, and density estimation. Journal of the American Statistical Association 97(458), 611–631 (2002)
12. Furey, T.S., Cristianini, N., Duffy, N., Bednarski, D.W., Schummer, M., Haussler, D.: Support vector machine classification and validation of cancer tissue samples using microarray expression data. BMC bioinformatics 16, 906–914 (2000)
13. Girolami, M., He, C.: Probability density estimation from optimally condensed data samples. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 1253–1264 (2003)
14. Guyon, I.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)

15. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46, 389–422 (2002)
16. Hastie, T., Tibshirani, R.: Discriminant adaptive nearest neighbor classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 18, 607–616 (1996)
17. Huang, C.J., Yang, D.X., Chuang, Y.T.: Application of wrapper approach and composite classifier to the stock trend prediction. Expert Systems with Applications 34(4), 2870–2878 (2008)
18. Koller, D., Sahami, M.: Toward optimal feature selection. In: Saitta, L. (ed.) Proceedings of the Thirteenth International Conference on Machine Learning (ICML), pp. 284–292. Morgan Kaufmann Publishers (1996)
19. Kononenko, I.: Estimating Attributes: Analysis and Extensions of RELIEF. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994)
20. Kwak, N., Choi, C.H.: Input feature selection by mutual information based on parzen window. IEEE Trans. Pattern Anal. Mach. Intell. 24, 1667–1671 (2002)
21. Liu, H., Setiono, R.: Feature selection via discretization. IEEE Transactions on Knowledge and Data Engineering 9, 642–645 (1997)
22. Narlikar, L., Hartemink, A.J.: Sequence features of dna binding sites reveal structural class of associated transcription factor. Bioinformatics 22(2), 157–163 (2006)
23. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer (August 2000)
24. Peng, Y.H.: A novel ensemble machine learning for robust microarray data classification. Computers in Biology and Medicine 36, 553–573 (2006)
25. Shakhnarovich, G., Darrell, T., Indyk, P. (eds.): Nearest-Neighbor Methods in Learning and Vision: Theory and Practice. MIT Press (2006)
26. Statnikov, A., Wang, L., Aliferis, C.F.: A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinformatics 9, 319–328 (2008)
27. Sun, Y.: Iterative relief for feature weighting: Algorithms, theories, and applications. IEEE Trans. Pattern Anal. Mach. Intell. 29(6), 1035–1051 (2007)
28. Sun, Y., Todorovic, S., Goodison, S.: Local-learning-based feature selection for high-dimensional data analysis. IEEE Trans. Pattern Anal. Mach. Intell. 32(9), 1610–1626 (2010)
29. Tao, Y., Vojislav, K.: Adaptive local hyperplane classification. Neurocomputing 71(13-15), 3001–3004 (2008)
30. Vincent, P., Bengio, Y.: K-local hyperplane and convex distance nearest neighbor algorithms. In: Advances in Neural Information Processing Systems, pp. 985–992. The MIT Press (2001)
31. Zhang, T., Oles, F.J.: Text categorization based on regularized linear classification methods. Information Retrieval 4(1), 5–31 (2001)

Towards Identity Disclosure Control in Private Hypergraph Publishing

Yidong Li[1] and Hong Shen[1,2]

[1] School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China
[2] School of Computer Science, University of Adelaide, SA, Australia
{ydli,hshen}@bjtu.edu.cn

Abstract. Identity disclosure control (IDC) on complex data has attracted increasing interest in security and database communities. Most existing work focuses on preventing identity disclosure in graphs that describes pairwise relations between data entities. Many data analysis applications need information about multi-relations among entities, which can be well represented with hypergraphs. However, the IDC problem has been little studied in publishing hypergraphs due to the diversity of hypergraph information which may expose to many types of background knowledge attacks. In this paper, we introduce a novel attack model with the properties of hyperedge rank as background knowledge, and formalize the rank-based hypergraph anonymization (RHA) problem. We propose an algorithm running in near-quadratic time on hypergraph size for rank anonymization which we show to be NP-hard, and in the meanwhile, maintaining data utility for community detection. We also show how to construct the hypergraph under the anonymized properties to protect a hypergraph from rank-based attacks. The performances of the methods have been validated by extensive experiments on real-world datasets. Our rank-based attack model and algorithms for rank anonymization and hypergraph construction are, to our best knowledge, the first systematic study for private hypergraph publishing.

Keywords: Identity disclosure control, Private hypergraph publishing, Anonymization, Community detection.

1 Introduction

Identity Disclosure Control (IDC) is a critical problem in private data publishing, and has been widely studied in previous work ([10,9,5,2,11]). Most of these studies focus on preventing entities from background knowledge attacks by modeling a social network as a graph [10,16]. However, the graph-based representation is neither sufficient nor realistic in real-world scenarios.

On the one hand, those potential data buyers (e.g. advertising agencies or application developer) are more interested in attributes reflecting the spending habit of an entity rather than the number of his/her friends. For example, the major purpose for a sport retailer paying for the data from Facebook is to figure

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 347–358, 2012.

out the entities who are members of a sport interest group. More important, such "interest group" data is usually real for an entity. We fabricate Bob as a member of Facebook, and assume that he is very conscious in protecting private information, such as the date of birth, the living place, the marriage status, and so on. Hence, it is almost unlikely to identify Bob, even the corresponding information is unique. However, as the inherent function and purpose of a social network, Bob will describe his real interests or join certain interest groups without any hesitation.

On the other hand, it is unrealistic to model background knowledge attacks on a social network with the graph-based representation in large-scale networks. For example, we take 1,000 students in Beijing Jiaotong University who have accounts in Renren.com, which is the most popular social network in China. With the graph-based representation, there are only 1.5% students who have unique degrees and 5.5% who have unique neighborhood substructures. However, by considering some properties of the interest groups, the unique rate can climb up to 33%.

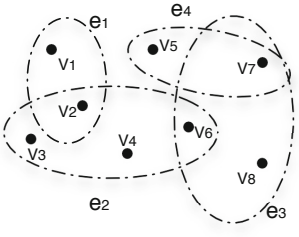


Fig. 1. An example of hypergraphs

In order to remedy the above issues, this paper proposes a hypergraph-based representation (seen in Figure 1) for a social network to depict a set of complex relational entities, such as grouping a population of entities with various attributes. A hypergraph-based representation is a mathematical construction that is quite useful to exploit relationships between different entities [17]. Generally, vertices represent entities and each hyperedge represents a relationship among a set of vertices.

Due to its specific structure, a hypergraph potentially faces more types of breaches with new background knowledge attacks. In our Facebook example, the members and their interest groups are natural to be modeled as a hypergraph and published to third parties without any privacy guarantee. Thus, an attack can be more valid (usually much easier) with the properties of such hypergraphs as background knowledge. For instance, Bob can be easily identified by others knowing his living habits. Notice that, the member-group relation can also be formed as a bipartite graph in our example, but a hypergraph is more general for our discussion, which is suitable for more complex relations such as tripartite

graphs [6]. In this paper, we discuss the IDC problem in social networks with a hypergraph-based representation.

1.1 Our Contributions

- discussing the IDC problem on hypergraphs by modeling rank-based attacks.
- formalizing a general model for *rank-based hypergraph anonymization*, and justifying the hardness of such a perturbation problem.
- proposing an efficient algorithm for rank anonymization, and exploring the issue of constructing a hypergraph with a specified rank set in the first place so far as we know.
- introducing the bias of communities as information loss incurred in hypergraph perturbation.

The remainder of this paper is organized as follows. Section 2 presents a brief survey of IDC on graphs. In Section 3, we model rank attack and introduce efficient metrics for data utility. Section 4 focuses on methods against the proposed rank attack and approaches for hypergraph construction. We present the experimental results in Section 5 and conclude this paper in Section 6.

2 Related Work

The IDC problem has been studied extensively on graphs. As pointed out in [2,9] simply removing the identifiers (or label) of the nodes does not always guarantee privacy. They study a spectrum of adversary external information and its power to re-identify individuals in a social network. The studies in [16,10] extend the above idea by modelling so called neighborhood attack and degree attack respectively. Specifically, the authors in [10] also proposed a two-step framework as property anonymization and graph construction, which is very useful to solve general anonymity problems on graphs.

Zheleva and Getoor [15] considered the problem of protecting sensitive relationships among the individuals in the anonymized social networks. This is closely related to the link-prediction problem that has been widely studied in the link mining community. The work in [14] studies how anonymization algorithms that are based on randomly adding and removing edges change certain graph properties.

Liu *et al.* [11] took weight into consideration for privacy preserving in social networks. They studied situations, such as in a business transaction network, in which weights are attached to network edges that are considered to be confidential. Then, they provide two perturbation strategies for this application. The research in [12] extend the above work by formulating an abstract model based on linear programming. However, the objective of their work still focuses on maintaining certain linear property of a social network by reassigning edge weights.

3 Problem Statement

3.1 Rank Attack

Let V denote a finite set of vertices, and let E be a family of subsets e of V such that $\cup_{e \in E} = V$. Then we call $G(V, E)$ a hypergraph with the vertex set V and hyperedge set E. A hyperedge e is said to be incident with a vertex v when $v \in e$. For a hyperedge $e \in E$, we use *rank* to denote the number of vertices in e, i.e., $r(e) = |e|$. For a vertex v, we use *rank sequence*, denoted as R, to represent the set of ranks of its incident edges $E(v) = \{e_1, e_2, \ldots, e_p\}$, i.e., $R = [r_1, r_2, \ldots, r_p]$. The set of rank sequences for all vertices in V is called the *rank set* of the hypergraph G, which is denoted by $\mathcal{R}_G = \{R_1, R_2, \ldots, R_n\}$, where n is the number of vertices in G.

Table 1. Tables for the example

(a) The adjacency matrix

	e_1	e_2	e_3	e_4
v_1	1	0	0	0
v_2	1	1	0	0
v_3	0	1	0	0
v_4	0	1	0	0
v_5	0	0	0	1
v_6	0	1	0	1
v_7	0	0	1	1
v_8	0	0	0	1

(b) Original $\mathcal{R}$

V	$\mathcal{R}$
v_1	[2]
v_2	[4, 2]
v_3	[4]
v_4	[4]
v_5	[2]
v_6	[4, 3]
v_7	[3, 2]
v_8	[3]

(c) 2-anonymity

V	$\mathcal{R}_2$
v_1	[3]
v_2	[4, 3]
v_3	[4]
v_4	[4]
v_5	[3, 2]
v_6	[4, 3]
v_7	[3, 2]
v_8	[3]

Specially, for a hypergraph being regular, all sequences in the rank set have the same dimension. Without loss of generality, we assume the elements in a rank sequence are sorted as in descending order, $r_1 \geq r_2 \geq \ldots \geq r_p$. Table 1b shows the rank set for the sample hypergraph. Now we model a potential attack on hypergraphs with the properties of edge rank as follows.

Definition 1. *(*rank attack*) Given a hypergraph $G(V, E)$, if the rank sequence R of a vertex $v \in V$ is unique in G, the vertex v can be identified from G by an adversary with the prior knowledge of R, even all vertices and hyperedges unlabelled.*

For example, in Figure 1, we have the corresponding adjacency matrix and the rank set in Table 1a and 1b for the sample. It shows that the vertices v_2, v_6, v_7 and v_8 have unique rank sequences, which have high disclosure risk with a rank attack. Let us take another real case as well: in our Facebook example, Bob is involved in three online interest groups, tennis, cooking and photography, the members in each group are $1,000$, 100 and 10. If the rank set $R = [1000, 100, 10]$ is unique in the hypergraph containing Bob, it is very likely to identify Bob from the published hypergraph.

3.2 Problem Definition

It is clear that the rank set of a hypergraph has to be investigated and modified (if necessary) before published, in order to protect from the above rank attack. We call this problem as *rank-based hypergraph anonymization (RHA)*. The initial idea is to generalize the values of the so-called quasi-identifier in a dataset, which is a group of attributes that can be uniquely identify individuals. In this paper, we define rank-based anonymity with the assumption that all attributes in a rank sequence form the quasi-identifier. However, all of the algorithms proposed later are suitable for the general case as well with a slight modification. Now, we first introduce the term of *rank anonymity* for a hypergraph as follows.

Definition 2. *(k-*rank anonymity*) A hypergraph $G(V, E)$ is k-rank anonymous if for every vertex v, there exist at least $k-1$ other vertices in the hypergraph with the same rank sequence as v.*

For example, from Table 1b, the vertices v_1, v_5, v_3 and v_4 are 2-rank anonymous, while others are all 1-rank anonymous. Therefore, the hypergraph shown in Figure 1 is 1-rank anonymous. By adding v_5 into the hyperedge e_1, the hypergraph G is 2-rank anonymous as shown in Table 1c.

Then, we formally define the RHA problem as follows.

Problem 1. (*rank-based hypergraph anonymization*) Given $\mathcal{R}_G$, the rank set of a hypergraph $G(V, E)$, and an integer k, construct a k-rank anonymous hypergraph $G^{'}(V, E^{'})$, such that the information loss $\mathcal{Z}$ is minimized.

In general, there exist two directions to solve the RHA problem: 1) changing the incident matrix of the hypergraph to adapt the requirement of the rank set, and 2) perturbing the rank set separately and then reconstructing a hypergraph with such modified rank set. The first method has the advantage of maintaining specified data utility globally while ensuring the security. However, such a technique is very inefficient to implement especially for very large hypergraphs. Hence, this paper will follow the second way to perturb a hypergraph, which is described as *rank anonymization (RA)* and *hypergraph construction (HC)* respectively.

3.3 Measuring Quality of Hypergraph Anonymization

Differentiating from some other problems, such as k-anonymity on transactional data, we use a conditional metric $\mathcal{Z} = (\mathcal{Z}; \mathcal{Z}_A)$ to assess the quality of an approach for RHA. The anonymizing cost $\mathcal{Z}_A$ is usually related to the operations of anonymization, while $\mathcal{Z}$ represents one of the important hypergraph property that we suppose to preserve. Here, we can only guarantee a solution of $\mathcal{Z}$ to be optimal for RHA with the condition of certain $\mathcal{Z}_A$. It can be seen as a trade-off between utility and efficiency. In other words, an anonymizing algorithm becomes too complex to implement with a real graph property as $\mathcal{Z}_A$, since it must construct the adjacency matrix to obtain the real property at each step of perturbation.

Anonymizing Cost. As our basic operations for perturbing are to add, delete or reallocate vertices in hyperedges, the method for rank anonymization is naturally required to minimize the changes of hyperedges. Given a hyperedge e and the anonymized $e^{'}$ correspondingly, we define the difference between their ranks as anonymizing cost for a hyperedge, i.e. $|r_e - r_{e'}|$. Then, given a rank set $\mathcal{R}_G$ of a hypergraph $G(V, E)$, we describe the total anonymizing cost as

$$\mathcal{Z}_A = \sum_{i=1}^{m} \sum_{j=1}^{g_i} \|R_{ij} - R_i^*\|^2 \tag{1}$$

where m is the number of anonymized groups, g_i and $R_i^* \in \mathcal{R}_G$ represent the number of objects and the anonymous object in each group.

Information Loss on Community Detection. As a hypergraph is powerful in representing the multi-relationship among vertices, an important and natural requirement is to detect communities in real-world applications [17,13]. Therefore, the methods for RHA also aim at minimizing the effect on community detection on hypergraphs published.

We use a popular metric, called modularity, which is known as a global quality function to identify communities. We revise the definition of modularity in [7] by using the terms of hypergraphs as the cumulative deviation from the random expectation.

$$\mathcal{M} = \sum_{g=1}^{N_C} \left(\frac{\sum_{\{v_i, v_j \in C_g \mid i \neq j\}} c_{ij}}{\sum_s r_s \left(r_s - 1\right)} - \frac{\sum_{\{v_i, v_j \in C_g \mid i \neq j\}} d_i d_j}{\left(\sum_s r_s\right)^2} \right), \tag{2}$$

where c_{ij} is the actual number of hyperedges in which i and j are together. Due to space limitations, the induction of Equation 2 is omitted here. Let $\mathcal{M}_G$ and $\mathcal{M}_{G'}$ be the modularity derived from G and $G^{'}$ respectively. Then, we can define the *modularity bias* as information loss,

$$\mathcal{Z}_M = \frac{|\mathcal{M}_G - \mathcal{M}_{G'}|}{\mathcal{M}_G}. \tag{3}$$

4 Algorithms

In this section, we propose algorithms for the RHA problem. It first states the hardness of RHA with anonymizing cost as the objective function, and introduces an efficient heuristic method based on the information loss defined in Equation 1. Then, we discuss hypergraph construction with a specified rank set, which is rarely mentioned in previous studies so far as we know.

4.1 Rank Anonymization

From the definition of RHA, it is obvious that rank anonymization, as the first step, is an optimization problem. The following theorem shows that such an optimization problem is NP-hard, even for the simplest case that all rank sets are with size 2.

Algorithm 1. The Rank Anonymization Algorithm

Input: A set of rank sets $\mathcal{R}$ and an integer k.
Output: An anonymized set $\mathcal{R}^{'}$.
1: **Initialization.**
1.1 find v_s and v_t in V with the most distance in $\mathcal{R}$;
1.3 form groups g_s and g_t containing v_s and v_t with their $k-1$ closest vertices respectively;
1.4 determine anonymous objects o_s and o_t and compute information loss for each group.
2: **Recursion**.
2.1 set all remaining vertices as 1-element group and initial the anonymous object as itself;
2.2 merge two groups with the lowest information loss;
2.3 re-calculate anonymous objects for each group;
2.4 go to 2.2 until every vertex is assigned to a group with size $[k, 2k)$.
3: **Perturbation**.
3.1 replace elements in each group by anonymous object;
3.2 merge all groups as $\mathcal{R}^{'}$ and return.

Theorem 1. *The optimal rank anonymization problem is NP-hard.*

Limited by space, we omit the formal proof (seen in the extended version of this paper). To guarantee the complexity in polynomial time, we introduce an efficient heuristic algorithm as a solution in Algorithm 1. This algorithm is similar with a family of data-oriented heuristics for microaggregation proposed in [3], while the major difference is the objective function due to anonymity.

The computational complexity of Algorithm 1 is $O(n^2 log\frac{n}{k})$. Here, we form a symmetric $n \times n$ distance matrix that each entry represents the Euclidean distance between two rank sets in $\mathcal{R}$. It reduces the complexity of the initialization step to linear. In each step of recursion, the algorithm introduce $O(n^2)$ operations to calculate all new distances among groups. Finally, there are $log\frac{n}{k}$ recursions due to the group merging. Therefore, the total complexity is $O(n^2 log\frac{n}{k})$.

4.2 Hypergraph Construction

Our next task for RHA is to reconstruct a hypergraph with a perturbed rank set $\mathcal{R}$. Some existing work [8,10] has studied popular construction techniques according to various properties of a graph, such as degree and spectrum. Unfortunately, most of these studies only consider graphs, and it has more concerns to apply the proposed methods on hypergraphs due to the computational complexity. The major reason is that one modification on an edge will affect a group of vertices rather than only two vertices in graphs. Specifically, there are two major challenges for the RHA problem: 1) the anonymized rank set has high possibility that is not realizable; and 2)the constructed hypergraph need to maintain the original community.

To explain the first challenge, we define the realizability of a rank set as follows.

Definition 3. *(*Realizability*) A rank set $\mathcal{R}$ is called realizable if and only if there exists at least one hypergraph $G(V, E)$ that has the exact same rank set with $\mathcal{R}$.*

This definition is extended from the realizability of degree on graph construction [4]. We state the following necessary and sufficient condition for a rank set to be realizable.

Lemma 1. *A rank set $\mathcal{R}$ is realizable if and only if, for any entry r_{ik} it holds αr_{ik} $(\alpha = 1, 2, \ldots)$ different vertices in $\mathcal{R}$ containing the same entry with r_{ik}.*

Proof. The sufficiency is obvious. As a hyperedge with rank r contains r vertices, there at least exists $\alpha = 1$. For the necessity, assuming a rank set has αr different vertices having an element with value r, it is easy to form α hyperedges with rank as r.

For example, in Table 1c, for $r = 2, 4, 3$, it holds $\alpha = 1, 1, 2$ respectively, and the 2-anonymized rank set is realizable. However, if we modify R_1 and R_8 to $[3, 2]$ and $[4]$, the rank set is still 2-anonymized but unrealizable with $\alpha = \frac{3}{2}, \frac{5}{4}, \frac{5}{3}$.

Apparently, an anonymized rank set $\mathcal{R}$ has very high probability that it is not realizable. Thus, Lemma 1 introduces a principle in how to develop a construction method for RHA to ensure the success of construction. The basic idea behind is to generate a realizable rank set from $\mathcal{R}$ based on Lemma 1 with minimal modification. Algorithm 2 takes a specified rank set $\mathcal{R}$ as inputs and returns a successfully constructed graph and an *approximate error* σ, which denotes the modification bias of the rank set. Steps from 3 to 9 describe a procedure to remove edges by matching each element in its rank set. Step 5 is a basic search to find all vertices containing the same rank with r_{ri}. Step 6 is crucial to modify a rank set to be realizable based on Lemma 1. Step 11 is to ensure the connectivity of the output hypergraph. If the algorithm terminates and outputs a hypergraph, then this hypergraph has the approximate specified rank set $\mathcal{R}$.

The computational complexity of Algorithm 2 is $O(n^2m^2)$, where n is the number of vertices and m is the maximal degree for all vertices. For each vertex v_i, there are maximal m hyperedges connecting v_i with other nodes. And for each hyperedge, the worst case is traversing all remaining vertices to find $S(v_r)$, which is $n \times m$ times. As there are n vertices, the total complexity is $O(n^2m^2)$.

In Algorithm 2, the basic operations are adding/deleting vertices in each hyperedge of the original graph to satisfy the privacy requirement. However, such an operation may affect the progress of finding communities. Thus, we provide a *community preserving procedure* aiming at minimizing the change of the community set $\mathcal{C}_G$. Our main idea is to first assign a two-way label for each vertex $v \in V$ in $G(V, E)$ according to the community and the min-cut that contain it. Then, we perform vertex addition or deletion only in its incident domain(s). For example, assuming that a hypergraph G has two non-overlapping communities C_1 and C_2 with the min-cut S, a label (C_1, S_v) for a vertex $v \in V$ implies $v \in C_1$

Algorithm 2. The Hypergraph Construction Algorithm

Input: A hypergraph $G(V, E)$ and an anonymized rank set $\mathcal{R}$.
Output: A hypergraph $G^{'}(V, E^{'})$ and the approximate error σ.
1: $V \leftarrow \{v_1, \ldots, v_n\}$, $E \leftarrow \emptyset$, $count \leftarrow 0$;
2: **while** $\mathcal{R}$ consists of non-zero elements **do**
3: pick a random vertex v_r with $R_r \neq \mathbf{0}$;
4: **for** $i \leftarrow 1$ to d_r **do**
5: find a set $S(v_r) := \{v_s \in V \mid \exists r_{sj} = r_{ri}\}$;
6: modify r_{ri} and all r_{sj} as $s \leftarrow |S(v_r)|$ in $S(v_r)$;
7: $\sigma \leftarrow \frac{|r_{ri}-s|}{r_{ri}}$;
8: form an edge e containing all vertices in $S(v_r)$;
9: $E \leftarrow E \cup e$, $r_{ri}, r_{sj} \leftarrow 0$;
10: $V \leftarrow V \cup v_r$;
11: amend the connectivity of $G^{'}$;
12: return $G^{'}(V, E^{'})$ and σ.

and $v \in S_v$. We also use S_0 as a virtual set to denote a vertex does not appear in any min-cuts. Therefore, in Algorithm 2, we can perform the selection of v_s within the domain where the elements have the same label. Apparently, this procedure is application-oriented since there exist a number of algorithms for community detection. However, this limitation can be released in the real-world applications, which the data publisher can make consistent standards on the methods of community detection with data users.

5 Experiments

The experiments are conducted on a 2.16 GHz Intel Core 2 Duo Mac with 4GB of 667MHz DDR2 SDRAM running the Macintosh OS X 10.5.8 operating system. All algorithms are implemented using Matlab 7.0.

We use three real-world datasets, named Mushroom, Nursery and Msweb, which contains 8,124, 12,960 and 32,711 vertices respectively, and 22, 8 and 294 attributes respectively. Specifically, each attribute takes only a small number of values, each corresponding to a specific category. In our experiments, we constructed a hypergraph for each dataset, where attribute values were regarded as hyperedges. Therefore, the Mushroom data includes 122 hyperedges, while Nursery and Msweb have 27 and 294 hyperedges respectively. All three datasets are from the UCI Machine Learning Repository [1].

5.1 Rank Attack on Real-World Data

Our first experiment is to show whether rank attack may happen on real-world datasets. We detect the possibility of rank attack on the test data with varying a specified parameter β, which is a threshold to assess a breach. That is, while the number of vertices sharing the same rank sequence is no larger than β, these vertices are recognized to be disclosed.

Table 2. Rank Attack on Real-World Data

	Disclosure rate (%)			
	$\beta = 1$	$\beta = 3$	$\beta = 5$	$\beta = 10$
Mushroom	16.18	20.24	38.36	55.14
Nursery	4.26	6.21	8.15	13.11
Msweb	4.55	7.59	10.61	13.64

Table 2 reports the percentage of vertices which can be successfully identified by rank attack. It clearly shows that the rank attack indeed be a real issue for hypergraph publishing. All testing datasets have relatively high risk of entity disclosure. For Mushroom data, the disclosure rate of rank attack with $\beta = 10$ is even high as 55.14%, which implies over half of its vertices can be uniquely identified. The rate is 4.55% with $\beta = 1$ for The Msweb contains around 600 individuals have high disclosure risk in the data with the rate 4.55% corresponding to $\beta = 1$. Also, the disclosure rate grows very quick as β increases. For example, the rate on the Nursery dataset increases nearly 10% with $\beta = 10$ than that with $\beta = 1$.

5.2 Impact on Anonymizing Cost $\mathcal{Z}_A$

In this section, we assess the cost incurred in applying various strategies for rank anonymization. As a comparison, we also implement a greedy anonymizing algorithm for rank attack, called GreedyRA.

The graphs in Figure 2 describe the relations between anonymizing cost $\mathcal{Z}_A$ and various k for the Mushroom, Nursery and Msweb datasets respectively. The results show that $\mathcal{Z}_A$ increases slowly while k is not large (e.g. $k < 20$) for both anonymizing algorithms. Furthermore, the GreedyRA arises much higher cost than the RA algorithm does in all cases as expectation. The biggest difference occurs in Msweb, which is over two times for every plot. In addition, the outcomes reveal the efficiency of GreedyRA with small values of k. For example, the costs by the two methods are very close to each other when $k < 50$ in the Nursery data. Usually, the indistinguish level is not required to be very high in the real-world applications. Thus, both anonymizing algorithms work efficiently in such cases.

5.3 Impact on Information Loss

The final experiment is to explore the relation between the modularity bias defined in Equation 2 and k. Figure 3 shows the relative changes of $\mathcal{Z}_M$with HCCP-RA, HC-RA and HC-RAG approaches by varying k. The modularity bias rises up as k increases that follows the similar trend of $\mathcal{Z}_H$. However, the gradients are not steep as that of $\mathcal{Z}_H$ especially when k is not large. This implies

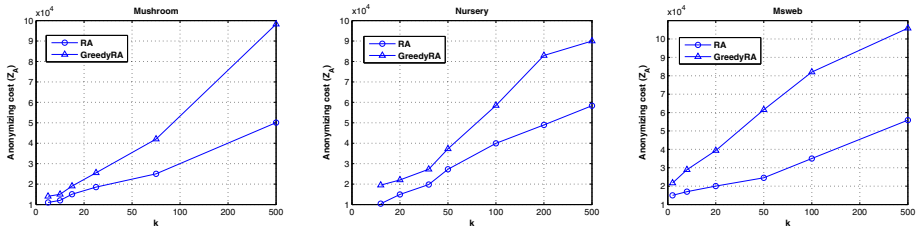


Fig. 2. The relation between $\mathcal{Z}_A$ and k

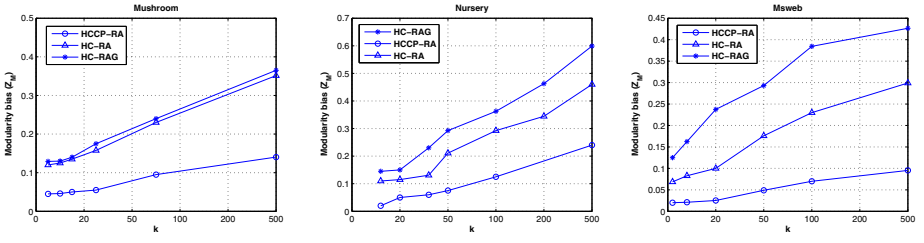


Fig. 3. The relation between $\mathcal{Z}_c$ and k

that the impact of perturbation on modularity is not significant as the intra-group error, as $\mathcal{Z}_M$ is a global measurement. Also, HCCP-RA shows much better performance than the others in all cases as expected. Moreover, the Mushroom data has an interesting result that HC-RA and HC-RAG produce very close plots with each other compared to the result of $\mathcal{Z}_H$. The reason is still not clear and we suppose it is related to certain structure properties of the dataset itself.

6 Conclusion

In this paper, we explored identity disclosure control in private hypergraph publishing. We addressed the problem of rank-based hypergraph anonymization by modeling a novel background knowledge attack with rank. We proposed an efficient heuristic algorithm for rank anonymization, which is shown NP-hard. We also studied the problem of constructing a hypergraph with a specified rank set, and provided methods maintaining the utility of community detection from the original hypergraph.

There are many issues of this work that need to be addressed in further research. As an NP-hard problem, it is worth to develop approximation algorithms for RHA. Also, it is worth to investigate how the proposed approaches affect other real hypergraph properties, such as diameter.

References

1. Asuncion, A., Newman, D.J.: UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences (2010)
2. Backstrom, L., Dwork, C., Kleinberg, J.: Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In: WWW 2007: Proceedings of the 16th International Conference on World Wide Web, pp. 181–190. ACM, New York (2007)
3. Domingo-ferrer, J.: Efficient multivariate data-oriented microaggregation. The VLDB Journal 15, 355–369 (2006)
4. Erdos, P., Gallai, T.: Graphs with prescribed degrees of vertices. Mat. Lapok 11, 264–274 (1960)
5. Feder, T., Nabar, S.U., Terzi, E.: Anonymizing graphs (2008)
6. Ghoshal, G., Zlatiić, V., Caldarelli, G., Newman, M.E.J.: Random hypergraphs and their applications. Phys. Rev. E 79(6), 066118 (2009)
7. Guimera, R., Sales-Pardo, M., Nunes Amaral, L.A.: Module identification in bipartite and directed networks. Physical Review E 76(036102) (2007)
8. Halbeisen, L., Hungerbuhler, N.: Reconstruction of weighted graphs by their spectrum. Eur. J. Comb. 21(5), 641–650 (2000)
9. Hay, M., Miklau, G., Jensen, D.: Anonymizing social networks. Technical Report 07-19, University of Massachusetts Amherst (March 2007)
10. Liu, K., Terzi, E.: Towards identity anonymization on graphs. In: SIGMOD 2008: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 93–106. ACM, New York (2008)
11. Liu, L., Wang, J., Liu, J., Zhang, J.: Privacy preservation in social networks with sensitive edge weights. In: 2009 SIAM International Conference on Data Mining (SDM 2009), Sparks, Nevada, pp. 954–965 (April 2009)
12. Egecioglu, O., Das, S., El Abbadi, A.: Anonymizing weighted social network graphs. In: The 26th International Conference on Data Engineering, ICDE 2010 (2010)
13. Vazquez, A.: Finding hypergraph communities: a bayesian approach and variational solution. Journal of Statistical Mechanics: Theory and Experiment (July 2009)
14. Ying, X., Wu, X.: Randomizing social networks: a spectrum preserving approach. In: SDM 2008: The SIAM International Conference on Data Mining, Atlanta, GA (April 2008)
15. Zheleva, E., Getoor, L.: Preserving the Privacy of Sensitive Relationships in Graph Data. In: Bonchi, F., Malin, B., Saygın, Y. (eds.) PInKDD 2007. LNCS, vol. 4890, pp. 153–171. Springer, Heidelberg (2008)
16. Zhou, B., Pei, J.: Preserving privacy in social networks against neighborhood attacks. In: ICDE 2008: The 24th International Conference on Data Engineering, pp. 506–515. IEEE Computer Society, Los Alamitos (2008)
17. Zhou, D., Huang, J., Scholkopf, B.: Learning with hypergraphs: Clustering, classification, and embedding. Advances in Neural Information Processing Systems 19, 1601–1608 (2007)

EWNI: Efficient Anonymization of Vulnerable Individuals in Social Networks

Frank Nagle[1], Lisa Singh[1], and Aris Gkoulalas-Divanis[2]

[1] Georgetown University, Washington, DC 20057, USA
[2] IBM Research-Zürich, Rüschlikon, CH-8803, Switzerland

Abstract. Social networks, patient networks, and email networks are all examples of graphs that can be studied to learn about information diffusion, community structure and different system processes; however, they are also all examples of graphs containing potentially sensitive information. While several anonymization techniques have been proposed for social network data publishing, they all apply the anonymization procedure on the entire graph. Instead, we propose a local anonymization algorithm that focuses on obscuring structurally important nodes that are not well anonymized, thereby reducing the cost of the overall anonymization procedure. Based on our experiments, we observe that we reduce the cost of anonymization by an order of magnitude while maintaining, and even improving, the accuracy of different graph centrality measures, e.g. degree and betweenness, when compared to another well known data publishing approach.

1 Introduction

Social networks, patient networks, email networks, and disease transmission networks are all examples of graphs that can be studied to learn about information diffusion, community structure and different system processes; however, they are also all examples of graphs containing potentially sensitive information. For some of these networks, it is not just the personal information that is sensitive, but also the position or existence of an individual in the graph. For example, the existence of a patient in a disease transmission network may be deemed as highly sensitive. As a result, a need exists to obscure sensitive topological information while still maintaining accurate graph properties for those studying these networks. Furthermore, because researchers are typically interested in data exploration applications, like community identification and information diffusion, our goal is to publish an anonymized, transformed network that is resilient against identity attacks and can be effectively studied as a graph.

A number of approaches have been proposed for anonymizing graphs. The research literature can be separated into two groups, those that add noise to the base graph and those that generalize the base graph. The former approaches use edge insertions and deletions to either deterministically create common patterns in the graph [24,16,25,3,26,7,6,22,19] or probabilistically add uncertainty [13,23,10,4]. The generalization strategy hides the detail level of the network by partitioning the graph into subgraphs, grouping nodes into clusters [5,20], or releasing specialized data structures that are specific for answering certain types of queries [12,14,11]. In this work, we investigate ways to publish a base graph that is sufficiently anonymized, contains sufficient detail for graph mining tasks using different graph properties, and is efficiently computed.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 359–370, 2012.

This work makes the following contributions: (1) Describes the use of simple graph metrics to guide the anonymization process; (2) Proposes a local anonymization strategy focusing on edge insertion or deletion to only the subset of nodes that are considered vulnerable and; (3) Experimentally evaluates the proposed method and shows that the released graph maintains a high degree of accuracy for different graph properties, including degree, betweenness, diameter, and average path length on five real world data sets, and are an order of magnitude more efficient than a well known approach that alters the entire graph.

The rest of the paper is organized as follows. Related literature is presented in section 2. In section 3, we provide necessary background and our privacy model. Our anonymization strategies are presented in section 4, followed by an experimental evaluation in section 5, and conclusions in section 6.

2 Related Literature

A number of previous works have shown that just removing the labels of published graphs is not sufficient for anonymization [2,12,17]. The main threats against these naïvely anonymized networks are *node re-identification* and *edge disclosures*.

Research focusing on adding noise to the base graph considers different strategies for edge insertion and edge deletion. Liu and Terzi [16] apply k-anonymity by ensuring that the degree of all nodes is k-anonymous. Zhou and Pei [25] focus on preventing neighborhood attacks by enforcing k-anonymous subgraphs based on a measure of the local neighborhood graph for all nodes. Their method relies on adding edges to the graph to make nodes that have distinct neighborhoods similar to other nodes. Wu et al [22] extend Zhou and Pei's work by introducing the k-symmetry model that accounts for anonymization based on the degrees of each node's neighbors. Similarly, Zou et al. [26] use k-automorphism to make subgraphs k-anonymous. More recently, Cheng et al. [6] developed the k-isomorphism method to preserve privacy at the subgraph level. Bhagat et al. [3] introduce the concept of *label lists* as a potential anonymity mechanism for obscuring the identity of a particular node. Zheleva and Getoor [24] present a number of anonymization strategies for avoiding sensitive edge inference breaches. Tai et al focus on a particular edge breach referred to as a friendship breach [19]. Das et al. [7] present a method for anonymizing social network graphs with weighted edges. Their linear programming anonymization method focuses on anonymizing the edge weights and preserving properties of the graph that are expressible as linear functions of the edge weights. In all these works, the anonymization procedure is applied to the entire graph. In contrast, we propose applying our anonymization procedure to a small subset of the full graph, thereby efficiently obtaining a releasable graph with comparable error.

3 Graph Structure and Privacy Model

3.1 Background

Let $G(V, E)$ represent a simple, undirected graph, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes and $E = \{e_{ij} = (v_i, v_j) | \, v_i \in V \text{ and } v_j \in V, \text{and } i \neq j\}$ is the set of

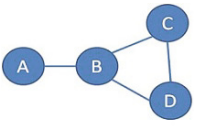


Fig. 1. A sample network graph

edges in G. Given a node v_i in V, its neighbors $N(v_i)$ are the set of vertices adjacent to v_i: $N(v_i) = \{v_j \mid (v_i, v_j) \in E,\ v_j \neq v_i,\ 1 \leq j \leq n\}$. Let $E(N(v_i))$ be the union of the edges between v_i and the nodes in $N(v_i)$ and the edges between the nodes in $N(v_i)$. Then the neighborhood subgraph of vertex v_i is $S(v_i) = \{V_S, E_S |\ V_S \in N(v_i) \cup v_i,\ E_S \in E(N(v_i))\}$.

The degree of a vertex is the size of the neighborhood, $|N(v_i)|$. Betweenness centrality is calculated by computing all of the graph's shortest paths and determining how many shortest paths a given node appears in. Sociologists measure the importance of individuals in a network using different centrality measures, including degree centrality (hubs) and betweeness centrality (brokers) [21]. Computer scientists have used these same measures in different graph mining algorithms.

3.2 Privacy Model

In this work, a data owner is interested in publishing a graph $G' = (V', E')$ that is an anonymized version of G. We assume that there is a bijective function $f : V \rightarrow V'$ that maps every vertex in V to a vertex in V'. We do not, however, require that all edges in E appear in E'. Some edges in E' may also not be present in E.

Most literature focuses on adversarial attacks on three parts of a graph: nodes, edges, and subgraphs. In this paper, our primary focus is on anonymizing nodes. We assume adversaries know the degree of one or more nodes, where the number of nodes known is small, $|V_{known}| << |V|$. However, the adversary does not know the neighborhood subgraph S of any of the nodes in V_{known}. Similar to related literature, k-degree anonymity occurs if at least k nodes have the same degree [16].

Definition 1. ***Node exposure*** *or a node identity breach occurs if any node in G is not k-degree anonymous.*

It is straightforward to determine if a node identity breach occurs using degree sets, where D_a is a set of vertices in G having degree a: $D_a = \{v_i | deg(v_i) = a \mid v_i \in V\}$. We define **D** as the set of all degree sets. For example, in Figure 1, the complete set **D** of degree sets is $D_1 = \{A\}$, $D_2 = \{C, D\}$, $D_3 = \{B\}$. If $k = 2$, then nodes A and B are exposed because D_1 and D_3 each have only one node in their sets.

Definition 2. ***Subgraph exposure*** *occurs if all the neighborhood subgraphs of nodes in a particular degree set D_j are the same.*

This results because the adversary knows $N(v_i)$, but not $S(v_i)$. However, if all the neighborhood subgraphs for a particular degree set are the same, then the adversary

will know $S(v_i)$ with certainty.[1] When we consider subgraph exposure, we must determine if all the subgraph neighborhoods are the same structure for a particular degree set. In other words, are they isomorphic? Determining if different neighborhoods in the graph are isomorphic is expensive to compute. However, because our neighborhood subgraphs are ego networks, we can use some simple social network metrics, e.g. clustering coefficient, to identify degree sets containing exposed neighborhood structures.

The clustering coefficient CC_{v_i} of a vertex v_i is a normalized value that shows how well connected the neighbors of vertex v_i are: $CC_{v_i} = \frac{2|S(v_i)|}{|N(v_i)|*(|N(v_i)|-1)}$ where $|N(v_i)| \geq 2$. The clustering coefficient of a node ranges from 0 (no neighbors connected to each other) to 1 (all neighbors connected to each other). In Figure 1, the clustering coefficient of B, C, and D are 0.333, 1 and 1, respectively.

To understand whether nodes in the same degree set D_a have the same or similar neighborhood structure, we can compare the variance of the clustering coefficient:

$$CC\_dif_a = \begin{cases} 0, if\ var(CC(D_a)) \leq \theta \\ 1, if\ var(CC(D_a)) > \theta \end{cases}$$

where $CC(D_a)$ is the clustering coefficient of each node in the degree set D_a, var is the variance of these values, and θ is the threshold for dissimilarity allowed in the neighborhood. If all the nodes in a degree set must have the exact same neighborhood subgraph to be exposed, then $\theta = 0$ and the exposure occurs with $var(CC\_dif_a) = 0$. However, if k is particularly large and want to extend the definition of subgraph exposure to allow for a very small percentage of the subgraphs for a degree set to not be isomorphic, then $\theta > 0$. Returning to our example in Figure 1, nodes C and D in D_2 have the same connectivity structure, $var(CC\_dif_a) = 0$; therefore, by definition both nodes have subgraph exposure.

Problem Statement: *Given a social network G, we want to publish G', a distorted version of G modified using a set of edge operations such that: 1) each vertex in V is represented in G'; 2) every vertex in V is k-degree anonymous in G'; 3) the degree of nodes that are already k-degree anonymous are not alterable; and 4) reasonable accuracy of different centrality and path measures exists in G'.*

While we focus on degree anonymity to quantify structural uniqueness in this work, any reasonable set of graph properties can be used to define unique parts of a graph, e.g. centrality measures, neighborhood measures, or subgraph structures. Therefore, we also propose a general definition for vulnerable components of a graph as follows:

Definition 3. *In a graph G, a **weak node**, v_w, is a node that is identifiable in the graph based on one or more graph properties. The presence of weak nodes in a graph reduces the overall anonymity of G. We define W as the set of all weak nodes in G. A **weak neighborhood subgraph**, S_{v_i}, occurs when the neighborhood subgraph of v_i is isomorphic to all the other neighborhood subgraphs in $D_{|N(v_i)|}$.*

[1] Note that our adversarial profile and definition of subgraph exposure differ from previous literature. Therefore, unlike previous literature, having the same neighborhood subgraphs leads to exposure.

Algorithm 1. EWNI - Anonymization Approach

1: **Input:** G, k, θ
2: **Output:** G'
3:
4: compute *degree_map*
5: $W = find\_weak\_nodes(G, k, degree\_map)$
6: $graph\_anonymization(W)$
7: $G' = graph\_construction()$
8: **return** G'

4 Anonymization Algorithms

In this section, we present our approach for graph anonymization. As described in Algorithm 1, our general approach is similar to that in [16]. The general approach proposed in [16] gathers the degrees of all nodes in G, identifies which degree sets do not have k nodes, and then changes the degree of those nodes to either match the closest degree that is k-degree anonymous or create a new degree set that is k-degree anonymous. After creating the new degree set, they construct a graph G' based on it.

Similarly, we begin by determining the set of weak nodes ($find\_weak\_nodes()$). We then apply a graph anonymization algorithm based on edge modification to only neighborhoods of weak nodes. We consider two strategies, edge insertion and probabilistic edge modification. After graph anonymization, a graph construction step follows that is based on the computed degree sequence. Here we follow the standard procedure proposed in Liu and Terzi [16] and will, therefore, focus on the first two parts of the task in the remainder of this section.

4.1 Finding Weak Nodes and Neighborhood Subgraphs

Our approach for calculating weak nodes is presented in Algorithm 2. For completeness, we also describe how to calculate weak subgraphs. Here, if $|D_i| < k$, a set of vulnerable nodes exists. Therefore, all nodes $v_i \in D_i$ in that degree set are added to a list of nodes (*weak_Da*) identified as weak due to their degree set length. We also track the difference between $|D_i|$ and k, allowing us to later rank the level of weakness of the nodes. When *CC_dif* $< \theta$, all nodes $v_i \in D_i$ are added to a list of node neighborhoods (*weak_cc*) identified as weak due to their *CC_dif* value. This is the $EWNI\_CC\_DIF()$ function in Algorithm 2. Running our weak node identification method returns two lists, one of weak nodes and one of nodes with weak neighborhoods. The EWNI algorithm has a time complexity of $O(|V| \cdot (|V| + |E|))$. However, in the worst case, it can require $O(|V|)$ more disk space. As will be illustrated in section 5, in practice we find that the number of weak nodes and nodes with weak neighborhood subgraphs in a network is a small proportion of the total number of nodes. It also remains small as the size of the network increases.

4.2 Anonymizing G

Our localized strategy focuses on changes to only the weak nodes and weak neighborhood subgraphs. The remaining parts of the graph remain the same. While we can our

Algorithm 2. Efficient Weak Node Identification

```
1: function find_weak_nodes(G, k, degree_map)
2:     i ← 1
3:     weak_cc ← {}
4:     weak_Da ← {}
5:     while (i <max_degree(G)) do
6:         D_i ← DEGREE_SET(i, degree_map)
7:         (cc_dif, weak_cc) ← EWNI_CC_DIF(D_i, weak_cc)
8:         if (|D_i| < k) then
9:             weak_Da ← weak_Da + D_i, k − |D_i|
10:    return (weak_cc, weak_Da)
11: function EWNI_CC_DIF(D_a, weak_cc)
12:    if (|D_a| = 0) then
13:        return (0,weak_cc)
14:    if (var(CC(D_a)) > 0) then
15:        return (1,weak_cc)
16:    else
17:        weak_cc ← weak_cc +D_a
18:        return (0,weak_cc)
```

weak node techniques to different anonymization algorithms, we choose to explain the proof of concept using the general framework proposed in [16]. We first describe our approach for anonymization that applies edge insertion to the weak nodes in the graph. We then explain a variation that considers both insertions and deletion.

Weak Node Edge Insertion: Let D_W be the degree sequence of the weak nodes. For this method, we directly apply a greedy algorithm similar to [16] to only weak nodes, W. This algorithm creates a group of the first k highest degree nodes that are not k-degree anonymous and assigns them all the highest degree in the group. The algorithm then computes two costs, the cost of merging the $(k+1)$-th node with the current group and the cost of starting a new group, where the cost is based on the number of edges that need to be inserted in each case. In order to help with the decision, the algorithm looks ahead to k other nodes. The algorithm continues recursively until all the weak nodes are considered. The run time is $O(|W| \cdot k)$. The proof of correctness is straightforward.

Probabilistic Weak Node Anonymity: For this method, we mimic the Weak Node Edge Insertion algorithm described above, but instead of always adding edges to make the k-sized group of nodes k-degree anonymous, we randomly determine whether to add or delete edges. Let p be the probability for deleting an edge and $q = 1 - p$ be the probability for adding an edge. We first sort the degrees of the nodes in W. Then given the sorted set of weak nodes, during each iteration for a size k group we randomly insert or delete edges to the nodes in each weak degree set based on p and q. If the decision is to insert an edge, we insert edges to the nodes in the group until all the degrees of the nodes are equal to the highest degree in the group. If the decision is to delete an edge, we delete edges and decrease the degrees in the group until all the degrees are equal to the lowest degree in the group. This process continues until all the weak degree

sets are members of k-degree anonymous degree sets. This method combines both a deterministic and probabilistic adding of noise to obtain k-degree anonymity.

Proof. (Sketch) Nodes that are not in D_W are already k-degree anonymous by definition. Since the nodes in D_W are considered in groups of k and, in each group, all nodes are assigned the same degree, these nodes become k-degree anonymous. Consequently, the produced graph (using [16]) satisfies k-degree anonymity as it consists of k-degree anonymous nodes.

Probabilistic Weak Neighborhood Subgraph Anonymity: While our focus is on node exposure, our probabilistic weak node anonymity algorithm could be extended for neighborhood subgraphs by only considering nodes and edges that are in W based on both $weak\_cc$ and $weak\_Da$. Generally, the algorithm would focus on adding and removing edges that exist between neighbors that are weak. We save this analysis for future work.

5 Experiments

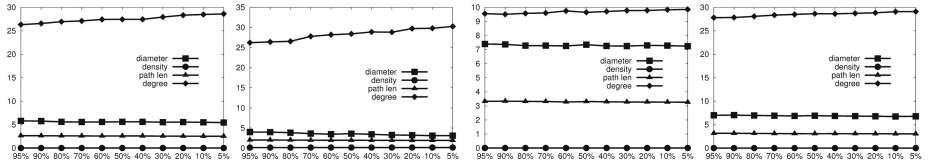
In this section we evaluate our approach in terms of 1) graph edit distance between G and G'; 2) accuracy of graph properties; and 3) efficiency of anonymization on five real world networks (graph properties shown in Table 1). The *PolBlogs* graph represents a network of hyperlinks between weblogs about US politics in 2005 [1]. The *Jazz* graph shows connections between different jazz musicians [8]. The *Email* graph is a network of email interchanges between members of the University Rovira i Virgili [9]. The *Wiki* graph is a network representing user participation in different elections [15]. Last, the *Facebook* graph is from a crawl of a subset of public Facebook pages. Because this crawl followed a snowball sampling protocol with multiple seeds, we remove nodes that only have a single degree since they are an artifact of the sampling approach.

Sensitivity Analysis of Probabilistic Anonymization: Before comparing our anonymization algorithm to other algorithms, we want to understand the sensitivity of the percentage of additions vs. deletions of edges. In other words, does the actual percentage of additions versus deletions affect the accuracy of the graph properties of interest?

Figure 2 shows the percentage of error introduced for our probabilistic method when we vary the percentage of edges that are inserted and deleted. Each experiment was run 10 times and the average results are presented. The x-axis shows the probability of deleting an edge as opposed to inserting it. The y-axis shows the amount of error introduced for each measure. This figure shows that the amount of error introduced is relatively constant, but does rise some as the probability of insertions becomes higher than deletions. Therefore, for the remainder of our experiments, we will set the probability of deleting an edge to 95% and the probability of adding an edge to 5%.

Table 1. Graph properties of data sets

Network Name	Nbr of Vertices	Nbr of Edges	Average Degree	Average Betweenness
Jazz	198	2742	27.7	121.65
Email	1133	5451	9.62	1475.01
PolBlogs	1222	16714	27.36	1060.76
Wiki	7115	100762	28.32	7884.65
Facebook	40531	157054	7.75	71262.98

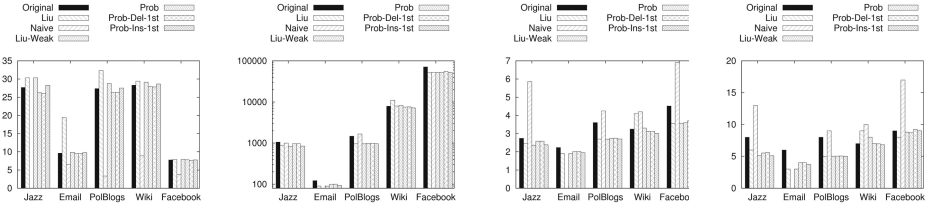


(a) Political Blogs (b) Jazz (c) Email (d) Wiki

Fig. 2. Graph properties percentage error as the probability of deletion decreases ($k = 3$)

Accuracy of Graph Properties: We now consider the accuracy of the released, perturbed graphs. The methods we consider are as follows: the original Liu and Terzi algorithm [Liu], removal of weak nodes from the graph [Naïve], Liu and Terzi applied to only weak nodes [Liu-Weak], our probabilistic anonymization [Prob], [Prob] with the first operation forced to be a deletion [Prob-Del-1st] or with the first operation forced to be an insertion [Prob-Ins-1st]. A naïve approach for anonymizing weak nodes is to produce a graph G' that simply removes each weak node, v_w in W and incident edges to nodes in $N(v_w)$ from G. In the last two methods, to reduce the variance of the basic probability anonymization procedure, we force the first operation to be consistent across runs. This is necessary because of the generally large variance in the degrees of the nodes in the first k weak nodes. Because they are the highest degree nodes, deleting or inserting has the largest amount of impact on this 1st group of nodes. Forcing the first action to always be the same reduces the variance to under 5%. The different algorithms were run ten times and the average error introduced by each method for $k = 10$ is shown in Figure 3. If a bar is missing, then G' was disconnected. We measured the error for varying values of k and the results were similar to those when $k = 10$. The x-axis shows the different data sets and the y-axis shows the value of the graph property as computed on the original graph G [Original], and on the anonymized graph G' produced by each tested method.

From Figure 3 we see that the best performer is [Prob-Del-1st] and the naïve removal of weak nodes generally results in the highest error across all data sets and measures. The exception to that is the betweenness calculation, where it generally outperforms the other methods. We suspect this occurs because the other approaches are adding at least a small fraction of edges to the graph, thereby creating new paths and potentially reducing the number of shortest paths each node lies on. With the exception of the political blogs data set, applying Liu and Terzi to just the weak nodes introduces less error than



(a) Average Degree (b) Average Betweenness (c) Average Path Length (d) Diameter

Fig. 3. Graph properties for different anonymization techniques ($k = 10$)

Table 2. Graph edit distance comparison to the algorithm of [Liu] with $k = 3$ and $k = 10$

	[Liu]	[Liu-Weak]	[Prob]
Jazz	0	-669, -13	-681, -324
Email	0	-84, -114	-131, -305
PolBlogs	0	-3529, -1242	-3529, -2547
Wiki	0	-11089, -6554	-11831, -10501
Facebook	0	-1138, -596	-2297, -6347

applying the algorithm to the entire graph. In general, our probabilistic methods perform comparable or better than the other methods on these data sets.

Since our methods do not guarantee a minimum number of edge modifications to G, we also compare the graph edit distance between G and G'. We set the baseline to be the Liu and Terzi method and compare the graph edit distances based on that method. Table 2 shows these results for each of the data sets averaged over 10 runs with $k = 3$ and $k = 10$, respectively. The table should be read as follows. For the Jazz dataset, the [Liu-Weak] algorithm needs 669 fewer graph edit operations than [Liu] when $k = 3$ and 13 fewer graph edit operations when $k = 10$. Looking at the entire table, we see that [Prob] has the smallest edit distance in all cases.

Finally, Singh and Zhan propose a measure called topological anonymity that uses node and subgraph exposure to quantify the level of risk associated with releasing a particular graph G' [18]. It is computed as follows:

$$ta = \frac{\sum_{i=1}^{\max(\deg(G))} \left[(|D_i| \times CC\_dif_i) - \begin{cases} 0, \text{if } |D_i| \geq k \\ |D_i|, \text{if } |D_i| < k \end{cases} \right]}{n}$$

where k represents the required number of nodes in a degree set and n is the number of nodes in G. The ta score, ranging from -1 to 1, with -1 indicating that the graph is highly susceptible to both node and neighborhood subgraph exposure, and 1 indicating that the nodes are well anonymized.

As another method to quantify the level of anonymity of G' compared to the other approaches, we compute the topological anonymity of G' for the different methods. Figure 4 (shown below) illustrates the improvement in the ta score after anonymization. All the algorithms improve with the exception of the naïve one.

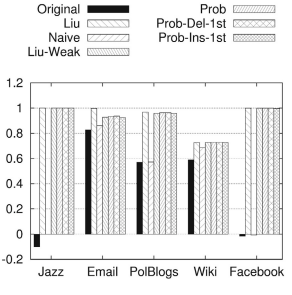


Fig. 4. Topological anonymity comparison

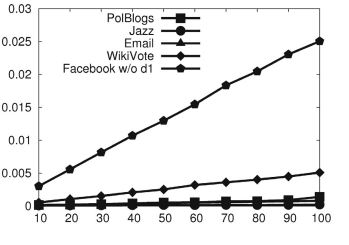


Fig. 5. Run times (seconds) as the percentage of weak nodes increases

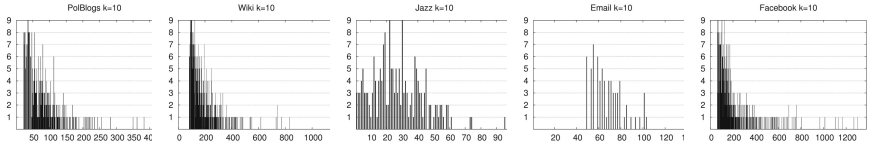


Fig. 6. Weak node distribution for $k = 10$

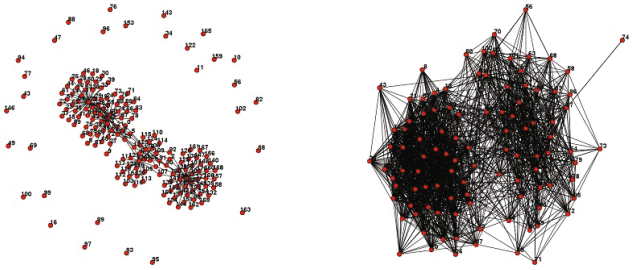
Weak Nodes Distribution: We now compare the distribution of weak nodes to determine if they are similar or different across our data sets. Figure 6 shows the distribution of weak nodes when $k = 10$. The x-axis shows the degree of each weak node, and the y-axis shows the number of nodes with that degree. In all cases, the maximum of the y-value is k-1, since degrees that have k or more nodes are not considered weak. These graphs show a number of interesting properties about the distribution of weak nodes. The main similarity among the graphs is that the far left side does not have any weak nodes, indicating that low degree nodes are generally k-degree anonymous. The Jazz network is the one exception. Second, we observe that as the degree increases, the number of nodes with that degree decreases and the bars become sparser as we move from left to right along the x-axis. This may be an indication that there are fewer nodes with high degree and that those nodes are not always weak. Both of these observations support theories that state social networks often follow a power law distribution. Finally, the figures highlight that the distribution of the weak nodes differs from data set to data set.

In addition to considering the number of weak nodes with each degree, we are also interested in the subgraphs formed by these weak nodes. They represent the portion of the graph that is most vulnerable to attack. Figure 7 shows that weak nodes tend to be highly connected, with all weak nodes in the Political Blogs network contained in one component, and the majority of weak nodes in the Facebook network contained in one component. We measure the vulnerability associated with subgraphs by considering their size and connectivity to other weak nodes. The *subgraph vulnerability index* is defined as: $SVI = \frac{|S_w|}{|W| \times |C_w|}$, where $|S_w|$ is the number of nodes in the weak subgraph(s), $|W|$ is the total number of weak nodes, and $|C_w|$ is the number of weak components. SVI is 1 and 0.822 for Political Blogs and Facebook, respectively.

Efficiency Results: Table 3 compares the run time of the Liu and Terzi algorithm to our delete first probabilistic anonymization algorithm with the probability of deleting

Table 3. Run Time Comparison (milliseconds)

Network Name	Preprocessing		Anonymization		Total	
	Liu	Weak Prob	Liu	Weak Prob	Liu	Weak Prob
Jazz	0.036	10.956	0.451	0.146	0.487	11.102
Email	0.062	25.239	8.423	0.089	8.485	25.328
PolBlogs	0.065	97.48	7.326	0.284	7.391	97.764
Wiki	0.25	1705.361	171.962	0.463	172.212	1075.824
Facebook without Degree 1	1.453	5187.103	4852	0.479	4853.453	5187.582
Facebook	8.484	73967.523	209784.52	1.125	209793.004	73968.648
Facebook doubled	17.217	155035.336	861238.351	5.277	861255.568	155040.613
Facebook quadrupled	33.781	319397.488	4392412.422	6.196	4392446.203	319403.684



(a) Facebook (b) Political Blogs

Fig. 7. Weak Subgraphs

an edge set to 95% and the probability of adding an edge set to 5% when $k = 10$. The second and third columns compare the preprocessing cost of the two approaches. The next two columns compare the anonymization approaches followed by the total run time in milliseconds.

Our preprocessing cost is higher than the original Liu and Terzi algorithm and is always the dominant cost of the approach. The Liu and Terzi algorithm precomputes a degree set map. Our algorithm precomputes a degree set map and clustering coefficients. While for small graphs our preprocessing cost is high, as the size of the graph increases, it increases linearly, resulting in an overall run time that is still less than the overall run time of the Liu and Terzi algorithm.

Table 3 shows that our anonymization run time increases sublinearly and is orders of magnitudes faster than Liu and Terzi as the size of the graph increases. This is because our cost is related to the number of weak nodes in the graph, which is a small fraction of the total number of nodes. To evaluate the run time of the anonymization algorithm as the number of weak nodes increases, we simulate an increase in the number of weak nodes for each data set. Figure 5 shows that the run time increases linearly. Therefore, even when the number of weak nodes increases, the algorithm performs efficiently.

6 Conclusions

This paper investigates anonymization of social graphs for data publishing. Current approaches apply anonymization techniques to the entire graph. We introduce the concept of weak nodes and propose approaches that only anonymize those nodes. We show that the number of weak nodes tends to be small in real world networks and anonymization focusing on these nodes is orders of magnitude faster and maintains the same level of accuracy and a low edit distance when compared to traditional methods. By not distributing the noise uniformly across the graph, more of the original distribution and properties are well maintained. Future work will investigate weak subgraph anonymization and try to understand the impact of releasing a partially generalized graph.

Acknowledgments. We would like to give a special thanks to our paper reviewers. Their insightful comments help improve the paper significantly. Finally, the experiments reported in this work were conducted at Georgetown University.

References

1. Adamic, L., Glance, N.: The political blogosphere and the 2004 US Election. In: WWW 2005 Workshop on the Weblogging Ecosystem (2005)
2. Backstrom, L., Dwork, C., Kleinberg, J.: Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography. In: WWW (2007)
3. Bhagat, S., Cormode, G., Krishnamurthy, B., Srivastava, D.: Class-based graph anonymization for social network data. In: VLDB (2009)
4. Bonchi, F., Gionis, A., Tassa, T.: Identity obfuscation in graphs through the information theoretic lens. In: ICDE (2011)
5. Campan, A., Truta, T.: Anonymization of centralized and distributed social networks by sequential clusterings. In: PinKDD (2008)
6. Cheng, J., Fu, A., Liu, J.: K-isomorphism: privacy preserving network publication against structural attacks. In: SIGMOD (2010)
7. Das, S., Egecioglu, O., Abbadi, A.: Anonymizing weighted social network graphs. In: ICDE (2010)
8. Gleiser, P., Danon, L.: List of edges of the network of jazz musicians. Adv. Complex Systems 6, 565 (2003)
9. Guimera, R., Danon, L., Diaz-Guilera, A., Giralt, F., Arenas, A.: Network of emal interchanges. Physical Review E 68 (2003)
10. Hanhijarvi, S., Garriga, G., Puolamaki, K.: Randomization techniques for graphs. In: SDM (2009)
11. Hay, M., Miklau, G., Jensen, D.: Enabling accurate analysis of private network data. Chapman & Hall, CRC Press (2010)
12. Hay, M., Miklau, G., Jensen, D., Towsley, D.: Resisting structural re-identification in anonymized social networks. In: VLDB (2008)
13. Hay, M., Miklau, G., Jensen, D., Weis, P., Srivastava, S.: Anonymizing social networks. Technical Report 19, University of Massachusetts (2007)
14. LeFevre, K., Terzi, E.: Grass: Graph structure summarization. In: SDM (2010)
15. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Signed networks in social media. In: ACM Conference on Human Factors in Computing Systems (2010)
16. Liu, K., Terzi, E.: Towards identity anonymization on graphs. In: SIGMOD (2008)
17. Narayanan, A., Shmatikov, V.: De-anonymizing social networks. In: IEEE Symposium on Security and Privacy (2009)
18. Singh, L., Zhan, J.: Measuring topological anonymity in social networks. In: IEEE Conference on Granular Computing (2007)
19. Tai, C.-H., Yu, P.S., Yang, D.-N., Chen, M.-S.: Privacy-preserving social network publication against friendship attacks. In: KDD (2011)
20. Tassa, T., Cohen, D.: Anonymization of centralized and distributed social networks by sequential clusterings. In: TKDE (2011)
21. Wasserman, S., Faust, K.: Social network analysis: methods and applications. Cambridge University Press, Cambridge (1994)
22. Wu, W., Xiao, Y., Wang, W., He, Z., Wang, Z.: k-symmetry model for identity anonymization in social networks. In: EDBT (2010)
23. Ying, X., Wu, X.: Randomizing social networks: a spectrum preserving approach. In: SDM (2008)
24. Zheleva, E., Getoor, L.: Preserving the privacy of sensitive relationships in graph data. In: KDD 2007 Workshop on Privacy, Security, and Trust (2007)
25. Zhou, B., Pei, J.: Preserving privacy in social networks against neighborhood attacks. In: ICDE (2008)
26. Zou, L., Chen, L., Ozsu, M.: KAutomorphism: A general framework for privacy preserving network publication. In: VLDB (2009)

A Pruning-Based Approach for Searching Precise and Generalized Region for Synthetic Minority Over-Sampling

Kamthorn Puntumapon and Kitsana Waiyamai

Department of Computer Engineering, Faculty of Engineering
Kasetsart University, Thailand, 10240
kamthorn.puntumapon@gmail.com, fengknw@ku.ac.th

Abstract. One solution to deal with class imbalance is to modify its class distribution. Synthetic over-sampling is a well-known method to modify class distribution by generating new synthetic minority data. Synthetic Minority Over-sampling TEchnique (SMOTE) is a state-of-the-art synthetic over-sampling algorithm that generates new synthetic data along the line between the minority data and their selected nearest neighbors. Advantages of SMOTE is to have decision regions larger and less specific to original data. However, its drawback is the over-generalization problem where synthetic data is generated into majority class region. Over-generalization leads to misclassify non-minority class region into minority class. To overcome the over-generalization problem, we propose an algorithm, called TRIM, to search for precise minority region while maintaining its generalization. TRIM iteratively filters out irrelevant majority data from the precise minority region. Output of the algorithm is the multiple set of seed minority data, and each individual set will be used for generating new synthetic data. Compared with state-of-the-art over-sampling algorithms, experimental results show significant performance improvement in terms of F-measure and AUC. This suggests over-generalization has a significant impact on the performance of the synthetic over-sampling method.

1 Introduction

Imbalanced data has been identified as one of ten most challenging problems in data mining [14]. A dataset is considered imbalanced when its class distribution is skewed. The skewed class distribution can be represented as a binary class, i.e., minority and majority class. The minority class is the one having a much smaller proportion of class examples, whereas the majority class contains a much higher proportion of class examples. In classification, we want to correctly classify on the both classes. However, most traditional classifiers are biased towards the larger number of examples. For example, the C4.5 decision tree algorithm [10] assumes data is from a well balanced class distribution. As a result, in cases of imbalanced class distribution, the algorithm is biased toward the majority class and treats the minority class as noise.

Solutions to the imbalanced class problem can be broadly divided into two categories [13]: algorithm level and data level. Data level solutions are independent of the

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 371–382, 2012.

classification algorithm; the preprocessed data can be used by any traditional classification methods. SMOTE [3] is a state-of-the-art of synthetic over-sampling algorithm that generates synthetic data along the line between minority data members and their selected nearest neighbors. The advantage of SMOTE is to have decision regions larger and less specific to the original data [9]. However, it suffers from over-generalization where synthetic data is generated into the majority class region. The reason is that SMOTE selects its neighbors without regard to the majority class. This problem leads to misclassifying non-minority class examples into minority class region.

Recently, many synthetic over-sampling methods [2–4, 7, 8] have been proposed to overcome the imbalanced class problem. Most of the proposed algorithms are based on SMOTE and directly select seed examples to generate new synthetic data. Borderline-SMOTE [7] directly selects seed examples based on the decision boundary. Borderline data is used as a seed example to generate new synthetic data. Safe-Level-SMOTE [2] directly positions synthetic data between two minority examples based on the number of minority data neighbors. However, these methods are not intended to solve the over-generalization problem. To the best of our knowledge, MSYN [4] is the first synthetic over-sampling method that tries to overcome over-generalization. MSYN uses 1-NN's margin to select a synthetic example. However, in case of a very small number of minority data, MSYN has the same problem as other methods as well.

In this paper, we propose an algorithm to overcome the over-generalization problem in imbalanced data. To avoid over-generalization, a greedy filtering strategy is employed to search for precise and generalized sets of minority data. Individually, precise and generalized sets are then used as seed sets to generate synthetic minority data. TRIM is a preprocessing algorithm for synthetic over-sampling methods. Therefore, it can be used as a preprocessor for most existing synthetic over-sampling methods. In this paper, TRIM is used as a preprocessing step to generate synthetic minority data for SMOTE, called TRIM-SMOTE. The experimental results show significant performance improvements in terms of F-measure and AUC over SMOTE. For F-measure, TRIM-SMOTE statistical significance outperforms SMOTE in 26 out of 33 experiments. For AUC, TRIM-SMOTE significance outperforms SMOTE in 22 out of 33 experiments.

2 Related Work

Studies using synthetic minority over-sampling [2–4, 7, 8] as a technique to balance class distribution in the literature are designed based on the Synthetic Minority Over-sampling TEchnique (SMOTE) [3]. SMOTE employs k-nearest neighbors (k-NN) as a range to couple two minority examples; new synthetic data is randomly generated along the line between the two minority examples. That is, SMOTE uses only minority data to generate new synthetic data without regard to the majority class. The way SMOTE generates new data can be interpreted as merging the two minority data into a disjunct with synthetic data. As a result, SMOTE makes decision regions larger and less specific to the original data. That is, the minority class is generalized and even over-generalized. There have been several other techniques to generate minority synthetic data; in this section we describe some significant works relevant to the work here.

Borderline-SMOTE (BSMOTE) [7] uses the basic assumption that data nearby the decision boundary has more chance to be misclassified than data far from the decision boundary. The author further proposed a criteria to identify borderline data based on k-NN; BSMOTE uses both minority and majority data in k-NN. The ratio of the number of neighborhood minority examples is used as criterion to identify importance of borderline data. Only borderline data and their neighbors are used to generate synthetic data. As a result, synthetic data is generated in the overlapping region between two classes. Therefore, borderline SMOTE also suffers from over-generalization. The algorithm may cause severe over-generalization due to the focus sampling in the overlap area.

Safe-Level-SMOTE (SSMOTE) [2] has been proposed to directly position synthetic examples instead of random positioning. The safe level criteria is based on the number of neighborhood minority data in k-NN. That is, the higher number of neighborhood minority data, the safer the position is. Since Safe-Level-SMOTE is based on SMOTE, new synthetic data are generate on a line between two minority examples. A new synthetic example is generated nearby a minority example that has higher number of neighborhood minority data.

To the best of our knowledge, Margin-guided Synthetic Over-sampling (MSYN) [4] is the first synthetic over-sampling method that claims to overcome over-generalization. MSYN uses 1-NN margin to estimate goodness of the synthetic data. The algorithm bias prefers synthetic data that has a large margin on both minority and majority classes. The synthetic data is ranked by the margin, MSYN then selects the top M values for use as synthetic minority data. MSYN trends to generate new synthetic data on a well separated region while avoiding the boundary region. Although MSYN can be used to avoid over-generalization, it uses all features to select a synthetic example. However, many existing classification methods usually use several good features to classify data. In this work, we propose a method to select precise seed sets on particular features. The proposed method searches one feature space at a time to filter out irrelevant data while maintaining seed data.

3 Methodology

The goal of the TRIM algorithm is to avoid over-generalization. The basic idea is to identify sets of minority data having the best compromise between generalization and precision. The algorithm starts with a single set containing all the data. For each feature, every splitting point is identified and evaluated based on the TRIM criteria, described below. A splitting point is identified by taking the midpoint between two adjacent sorted values having different classes. The best splitting point will be used to split the data into two sets, i.e., left and right sets. Iteratively, each set of minority data is split into smaller sets until the stopping criterion is satisfied. The final sets are minority datasets are used as seed data to generate synthetic data via SMOTE, namely TRIM-SMOTE.

3.1 TRIM Criteria

To avoid over-generalization, the seed data should be precise while maintaining their generalization. Therefore, the TRIM criteria, Equation (1), is used as a measure of

precision and generalization. The higher the $TRIM$ value, the more precise and generalized the seed data.

$$TRIM = \frac{|minority|^2}{N} \tag{1}$$

where $|minority|$ is the size of the minority data. To get better seed data, TRIM Gain ($T-Gain$) is evaluated against two splitting datasets, i.e, left and right sets. $T-Gain$ is compared to $TRIM$. If $T-Gain > TRIM$, a better seed data is obtained by a binary splitting operation. Define $|minority_{left}|$ and $|minority_{right}|$ as the number of minority data in the left and right subsets; let N be the total number of examples, N_{left}, and N_{right} be the number of data in the left and right subsets. With this notation,, $T-Gain$ can be calculated as

$$T-Gain = max(\frac{|minority_{left}|^2}{N_{left}}, \frac{|minority_{right}|^2}{N_{right}}) \tag{2}$$

The equation (2) is designed to capture two characteristics of SMOTE. The first characteristic is to generate new synthetic data from a couple of minority examples. Therefore, only minority data is used to evaluate precision. The second characteristic is synthetic data is always generated within the convex hull of the minority data. Therefore, another objective of $T-Gain$ is to identify irrelevant majority data located outside the convex hull and filter them out.

Fig. 1 shows a dataset containing two classes. Minority examples are shown as diamonds, and majority examples are stars. The convex hull of the minority class is illustrated by a dashed line. The relevant majority data is A,B,C,D, and E. The rest of the majority data are irrelevant. The seed data lie within the solid line. Similar to decision trees, seed data are modeled with a hyper-rectangle. Thus, the irrelevant majority data that locate inside the solid line will be mis-identified as relevant majority data.

The standard maximum function is used in Equation 2 to focus on improvement of the seed set. The first objective is to filter out irrelevant majority data located outside the convex hull of the minority data. We want to focus on improvement of seed data while ignore irrelevant data. The second objective is to identify which set of data is splitting seed data. To obtain a more precise seed data, some minority examples are split from the old seed data. The max is used to detect which of the subsets to use as splitting seed data. The splitting seed data is then compared against the old seed data to evaluate the improvement.

3.2 TRIM Algorithm

The TRIM algorithm uses a greedy approach to search for set of minority data while filtering out irrelevant data. Although this approach does not guarantee finding the global optimum, it provides a good approximation to the optimal set. The following pseudocode describes the algorithm.

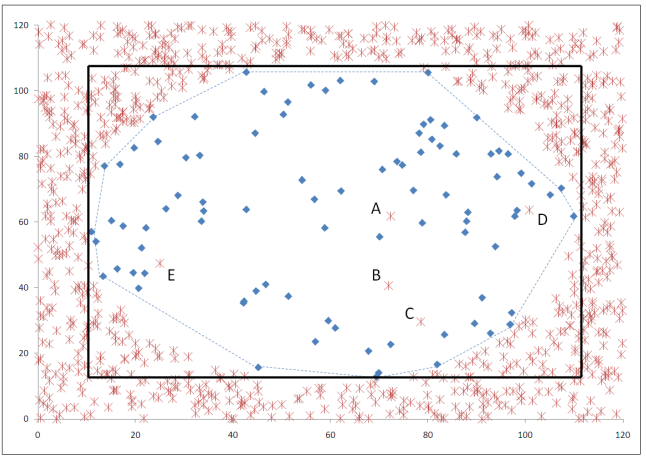


Fig. 1. The artificial dataset with its convex hull and seed set boundary after processing by TRIM

Algorithm TRIM(D)
Input: data D
Output: list $Seed$

1. add D to $Leaf$
2. While $Leaf$ and $Candidate$ is not empty{
3. For each $Leaf$ {
4. Initialize all splitting points S on $Leaf_i$
5. calculate $TRIM$ on $Leaf_i$
6. For each S {
7. accept the highest $T - Gain_{max}$ that does not split any minority data at S_{max} splitting point
8. }
9. if $(T - Gain_{max} > TRIM)$ {
10. split $Leaf_i$ into $Leaf_{left}$ and $Leaf_{right}$ using S_{max}
11. if ($subset_{left}$ contains any minority data)
12. add $Leaf_{left}$ to $Leaf$
13. if ($subset_{right}$ contains any minority data)
14. add $Leaf_{right}$ to $Leaf$
15. remove $Leaf_i$ from $Leaf$
16. } else {
17. add $Leaf_i$ to $Candidate$
18. remove $Leaf_i$ from $Leaf$
19. }
20. }
21. For each $Candidate$ {
22. Initialize all splitting points S on $Candidate_i$
23. calculate $TRIM$ on $Candidate_i$

24. For each S {
25. accept highest $T - Gain_{max}$ at S_{max} splitting point
26. }
27. if ($T - Gain_{max} > TRIM$) {
28. split $Candidate_i$ into $Candidate_{left}$ and $Candidate_{right}$ using S_{max}
29. if ($Candidate_{left}$ contains any minority data)
30. add $Candidate_{left}$ to $Leaf$
31. if ($Candidate_{right}$ contains any minority data)
32. add $Candidate_{right}$ to $Leaf$
33. remove $Candidate_i$ from $Candidate$
34. } else {
35. add $Candidate_i$ to $Seed$
36. remove $Candidate_i$ from $Candidate$
37. }
38. }
39. }
40. return $Seed$

In line 1, the algorithm starts with a single set containing all data. The algorithm consists of two main steps which are irrelevant data filtering and candidate splitting. The first step, irrelevant data filtering, is shown in lines 3-20. The main objective of this step is to filter out irrelevant majority data. The splitting point will be used with a constraint that is no minority data will be split into different set. This constraint ensures only irrelevant majority data is split in this step and the relevant majority data is used to splits minority data in the next step. Candidate splitting is performed in lines 21-39. This step focuses on splitting some minority data to get more precise seed data. Both split minority data and new seed data will be used in the 1st step of the next iteration. This process iterates until no more better splitting point is found.

In lines 4-5 of the irrelevant data filtering step, the algorithm starts by initializing available splitting points and $TRIM$ value of the whole dataset. In lines 6-8, $T - Gain$ is evaluated on every splitting point that splits all minority data into one set. In lines 10-15, the data is split into two sets (left and right). All majority data will be filtered out if they do not contain any minority examples. If no irrelevant data is found, the data is sent to the second step at line 17.

In candidate splitting, the algorithm is similar to the filtering step. However, this step focuses only on minority data, as shown in line 25. Notice that each seed data will be processed in both steps at least once. In order to maintain the generalization, the algorithm does not split minority data from majority data that are located outside the convex hull. To ensure this condition, all the irrelevant majority data are filtered out in the filtering step. Thus, the second step will split minority data based only on relevant majority and minority data.

The higher the $T - Gain$ in Equation (2), the more precise and generalized data is obtained. In line 35, seed data will be generated when $T - Gain \leq TRIM$. This can be interpreted as meaning that no more precise and generalized data can be found. Although seed data is well-approximated, some seed data can be considered as noise. To filter out noise, a threshold minimum precision ($minPrecision$) is used to select

seed data. Define $|minority_i|$ as the number of minority data in $Seed_i$; N_i be the total number of data in $Seed_i$. The precision of $Seed_i$ ($precision_i$) can be expressed mathematically as shown in Equation (3).

$$precision_i = \frac{|minority_i|}{N_i} \tag{3}$$

The precision value of a seed is compared against $minPrecision$. The seed set will be filtered out if $precision < minPrecision$. Intuitively, if we set $minPrecision$ too high, we lose some information. However, if we set $minPrecision$ too low, noise can happen in the seed data. We experimentally selected $minPrecision = 0.3$ as suitable to preserve information while filtering out noise.

The TRIM algorithm calculates $T - Gain$ on every splitting point. Define M as the number of attributes and N be number of data values. The maximum number of splitting points is $N - 1$, and the maximum number of iterations is $N - 1$ in the case where only one example data is split at each iteration. Hence, the time complexity of the algorithm is $O(MN^2)$. However, in experiments the critical step was found to be data sorting, having complexity $O(MNlog(N))$.

4 Experimental Results

In the following, the results of TRIM-SMOTE, SMOTE, and MSYN are compared using 11 continuous datasets. All experiments were conducted using 10 random seeds on 10-fold cross validation with three levels of sampling, i.e., 100%, 300%, 500%. Two measurements, AUC [1] and F-measure [12], are used to evaluate performance. Each algorithm was used in at least 3,300 experiments. The experiments use WEKA's C4.5 [6] as a classification model with default configuration, i.e., weka.classifiers.trees.J48 -C 0.25 -M 2. For TRIM, $minPrecision$ was set to 0.3; parameters for the other algorithms were set exactly same as in their papers.

Table 1. Experimental datasets

#	Dataset	#Attributes	%Minority	#Minority	Data size
1	letter-A	16	3.94	789	20000
2	arrhythmia-6	280	5.53	25	452
3	libras-10	90	6.67	24	360
4	glass-3	10	7.9	17	214
5	mfeat-fourier1	76	10.0	200	2000
6	yeast-ME3	9	10.9	162	1484
7	breastTissue-fad	10	14.15	15	106
8	segment-path	19	14.28	330	2310
9	bloodTransfer	4	23.7	178	748
10	haberman	3	26.4	81	306
11	ionoshpere	35	35.8	126	351

In Table 1 the 11 UCI datasets [5] are sorted by percentage of minority class. The percentages vary from 3.9% to 35.8%. Each dataset has five attributes, i.e., dataset's name, number of attributes (#Attributes), percentage of minority data (%Minority), number of minority data (#Minority), and size of dataset (Data size). All datasets have binary class, i.e, minority class and majority class. However, the 1st - 8th datasets have more than two classes. Therefore, we used one class as the minority class, and grouped the others as a majority class. For example, yeast-ME3 contains two classes, i.e., ME3 and others. The ME3 and others are considered as minority and majority classes, respectively.

To illustrate the impact of over-generalization, we examine results for the the segment-path and ionosphere datasets in term of AUC and F-Measure.

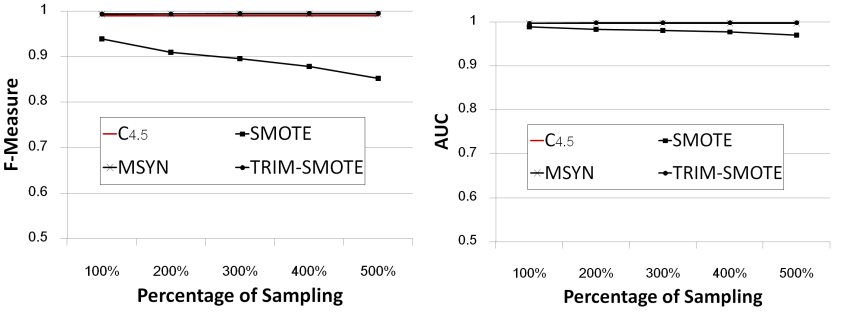


Fig. 2. The negative impact of over-generalization in term of F-Measure and AUC on segment-path dataset

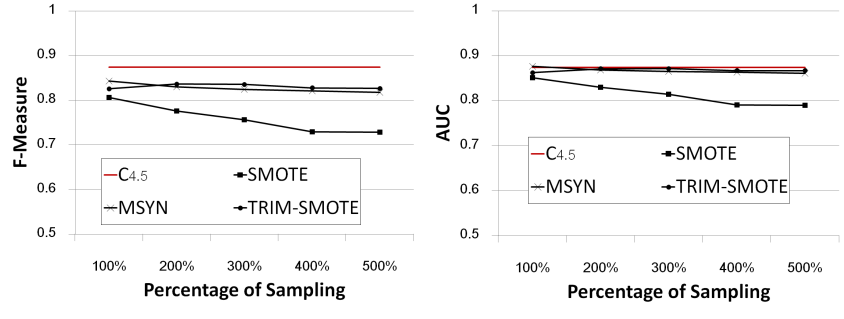


Fig. 3. The negative impact of over-generalization in term of F-Measure and AUC on ionoshpere dataset

As shown in Fig. 2, the three algorithms (decision tree C4.5, TRIM-SMOTE, and MSYN) exhibit comparable performance with respect to both AUC and F-measure. This can be interpretted to mean that performance improvement is not guaranteed by synthetic over-sampling methods. However, SMOTE is degraded on every percentage of sampling and its performance is worse when the sampling percentage increases. The reason is that MSYN and TRIM are designed to handle over-generalization while SMOTE suffers from this problem.

In Fig. 3, every synthetic over-sampling method, i.e., SMOTE, MSYN, and TRIM-SMOTE yields lower performance with respect to F-measure when compared to C4.5. However, we obtained stable performance with MSYN and TRIM-SMOTE, i.e., 82.8% F-measure with 0.7% standardization. In contrast, SMOTE always yields lower performance on every increasing sampling percentage. In terms of AUC, decision tree C4.5, TRIM-SMOTE, and MSYN show comparable performance while SMOTE is always lower. These four graphs illustrate the negative impact of over-generalization.

Table 2. Result in terms of F-measure in the experiments performs on eleven UCI datasets. A value will be underlined when the particular algorithm yield the highest F-measure among all four algorithms.

Dataset	%Minority	#Minority	%Sampling	C4.5	SMOTE	MSYN	TRIM-SMOTE
letter-A	3.94	16	100%	94.7%	93.9%	94.3%	95.2%
			300%	94.7%	92.3%	93.7%	95.2%
			500%	94.7%	91.8%	93.7%	94.8%
arrhythmia-6	5.53	25	100%	62.7%	58.1%	58.1%	64.3%
			300%	62.7%	40.8%	40.9%	64.1%
			500%	62.7%	36.2%	36.5%	69.6%
libras-10	6.67	24	100%	57.1%	56.6%	56.2%	57.0%
			300%	57.1%	52.7%	59.0%	60.3%
			500%	57.1%	50.8%	56.8%	65.6%
glass-3	7.9	17	100%	51.6%	35.0%	42.5%	47.7%
			300%	51.6%	30.3%	39.8%	47.9%
			500%	51.6%	29.6%	37.4%	43.8%
mfeat-fourier1	10.0	200	100%	97.4%	89.3%	98.1%	96.7%
			300%	97.4%	79.7%	98.3%	96.9%
			500%	97.4%	75.7%	98.3%	96.7%
yeast-ME3	10.9	162	100%	76.3%	76.3%	77.3%	77.4%
			300%	76.3%	74.9%	76.0%	76.3%
			500%	76.3%	75.0%	76.5%	78.4%
breastTissue-fad	14.15	15	100%	41.6%	26.7%	38.2%	36.9%
			300%	41.6%	34.8%	36.9%	39.4%
			500%	41.6%	33.2%	34.7%	42.0%
segment-path	14.28	330	100%	99.2%	93.8%	99.2%	99.2%
			300%	99.2%	89.5%	99.2%	99.3%
			500%	99.2%	85.1%	99.2%	99.3%
bloodTransfer	23.7	178	100%	46.9%	47.9%	48.3%	48.5%
			300%	46.9%	47.4%	47.6%	48.6%
			500%	46.9%	47.7%	47.0%	47.8%
haberman	26.4	81	100%	41.1%	49.5%	48.1%	51.1%
			300%	41.1%	47.9%	50.1%	48.2%
			500%	41.1%	45.6%	49.3%	48.1%
ionoshpere	35.8	126	100%	83.3%	80.5%	84.2%	82.5%
			300%	83.3%	75.5%	82.3%	83.4%
			500%	83.3%	72.8%	81.7%	82.5%
Win/Draw/Lose Significant				17/7/9	26/0/7	17/6/10	NA
1st rank				9	0	7	20

Performance of C4.5 (without sampling), SMOTE, MSYN, and TRIM-SMOTE on the eleven datasets in terms of F-measure and AUC are shown in Table 2 and 3. C4.5 is used as a baseline to evaluate improvement or decreased performance of the three algorithms. The bottom rows provide a comparison of each of the three algorithms with TRIM-SMOTE. The row labeled (Win/Draw/Lose Significant) summarizes the number of cases where the algorithm significant outperforms, equals, or performs worse than TRIM-SMOTE. To evaluate statistical significant, Wilcoxon signed rank test [11]

Table 3. Result in terms of AUC in the experiments performs on eleven UCI datasets. A value will be underlined when the particular algorithm yield the highest AUC among all four algorithms.

Dataset	%Minority	#Minority	%Sampling	C4.5	SMOTE	MSYN	TRIM-SMOTE
letter-A	3.94	16	100%	96.5%	96.8%	96.2%	97.3%
			300%	96.5%	96.3%	96.1%	97.5%
			500%	96.5%	96.2%	96.3%	97.3%
arrhythmia-6	5.53	25	100%	80.8%	82.7%	82.7%	81.8%
			300%	80.8%	73.7%	73.6%	82.2%
			500%	80.8%	71.8%	72.0%	86.5%
libras-10	6.67	24	100%	77.5%	79.8%	77.2%	78.0%
			300%	77.5%	80.5%	80.5%	78.0%
			500%	77.5%	83.0%	79.3%	78.3%
glass-3	7.9	17	100%	72.0%	64.2%	68.5%	72.4%
			300%	72.0%	64.3%	68.5%	71.8%
			500%	72.0%	67.9%	66.9%	70.4%
mfeat-fourier1	10.0	200	100%	98.3%	98.0%	98.8%	98.0%
			300%	98.3%	96.7%	99.1%	98.2%
			500%	98.3%	96.1%	99.2%	98.2%
yeast-ME3	10.9	162	100%	87.0%	89.4%	88.4%	89.0%
			300%	87.0%	90.3%	87.8%	89.5%
			500%	87.0%	91.4%	88.1%	90.8%
breastTissue-fad	14.15	15	100%	64.4%	57.8%	65.9%	64.5%
			300%	64.4%	65.8%	64.3%	65.7%
			500%	64.4%	65.4%	63.2%	67.7%
segment-path	14.28	330	100%	99.6%	98.8%	99.6%	99.6%
			300%	99.6%	97.9%	99.6%	99.7%
			500%	99.6%	96.9%	99.6%	99.7%
bloodTransfer	23.7	178	100%	65.2%	65.8%	66.1%	66.2%
			300%	65.2%	65.4%	65.6%	66.3%
			500%	65.2%	65.7%	65.2%	65.7%
haberman	26.4	81	100%	61.0%	65.4%	64.7%	66.6%
			300%	61.0%	63.8%	66.0%	64.2%
			500%	61.0%	61.6%	65.3%	64.0%
ionoshpere	35.8	126	100%	86.5%	85.1%	87.6%	86.2%
			300%	86.5%	81.3%	86.4%	87.1%
			500%	86.5%	78.9%	86.0%	86.6%
Win/Draw/Lose Significant				18/2/13	22/3/8	13/5/15	NA
1st rank				8	4	8	15

with p-value greater than or equal to 95% evaluates on every cross validation result. The second row shows number of cases where the algorithm obtains the highest performance among all four algorithms. The result is underlined when it obtains the highest performance.

Table 2 shows the experimental results evaluated using F-measure. The table shows that TRIM-SMOTE provided the best classification in 20 of 33 cases, and was almost similar to C4.5. The table shows that SMOTE generally underperformed both C4.5 and TRIM-SMOTE. For datasets having a very small number of minority examples, namely, arrhythmia-6, libras-10, glass-3, and breastTissue-fad, SMOTE and MSYN show lower performance than C4.5, whereas TRIM-SMOTE is most similar to C4.5, showing more stable performance.

Table 3 shows experimental results evaluated using AUC. The table shows that SMOTE generally underperformed both C4.5 and TRIM-SMOTE. For the datasets having a very small number of minority examples, namely, arrhythmia-6, glass-3, and breastTissue-fad, SMOTE and MSYN show lower performance than C4.5, whereas TRIM-SMOTE is most similar to C4.5, showing more stable performance. For libras-10 and yeast-ME3, SMOTE shows highest performance in terms of AUC, while TRIM-SMOTE yielded the highest F-measure. This can be explained by the fact that SMOTE randomly generates synthetic data on every minority data without regarding to majority data, whereas TRIM-SMOTE tries to maintain a precise and generalized set of seed data. As a result, TRIM-SMOTE produces comparable recall but higher precision while SMOTE gain higher recall but lower precision.

These experimental results indicate that, of the four algorithms, SMOTE underperforms C4.5 the most, and its performance declines as the sampling percentage increases. We observed stable performance for MSYN on large datasets but less than C4.5 for small datasets.

5 Conclusions

Synthetic minority over-sampling is a method that generates new minority examples to balance an imbalanced class distribution. The advantage is that synthetic data does not duplicate the original minority data. Therefore, the classification model is not overfitted to synthetic data. Synthetic over-sampling has its own drawback: over-generalization. The problem is that the majority class region is confounded with synthetic minority examples. To overcome over-generalization, we propose an algorithm called TRIM as a preprocessing for synthetic over-sampling. TRIM searches for a set of precise minority examples while maintaining their generalization. The precise seed set is used as an input to a synthetic minority over-sampling method. Thus, TRIM can be used as a preprocessing algorithm for many available synthetic over-sampling techniques such as SMOTE, BSMOTE, and MSYN. Prior to sampling, evaluation or interpretation of the seed data can also be conducted. Empirical results also show encouraging improvement over SMOTE and MSYN. TRIM-SMOTE is able to cope with the over-generalization problem more than MSYN. Its performance is stable on large dataset, and increased on small datasets. Experiments on multi-class and multivariate datasets are to be explored in the future study.

References

[1] Bradley, A.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30(7), 1145–1159 (1997)

[2] Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C.: Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 475–482. Springer, Heidelberg (2009), http://dblp.uni-trier.de/db/conf/pakdd/pakdd2009.html

[3] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)

[4] Fan, X., Tang, K., Weise, T.: Margin-Based Over-Sampling Method for Learning from Imbalanced Datasets. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part II. LNCS, vol. 6635, pp. 309–320. Springer, Heidelberg (2011)

[5] Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml

[6] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations 11(1), 10–18 (2009)

[7] Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005, Part I. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005)

[8] He, H., Bai, Y., Garcia, E.A., Li, S.: Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: IJCNN, pp. 1322–1328. IEEE (2008), http://dblp.uni-trier.de/db/conf/ijcnn/ijcnn2008.html

[9] He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21, 1263–1284 (2009)

[10] Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)

[11] Ramsey, P.H., Hodges, J.L., Shaffer, J.P.: Significance probabilities of the wilcoxon signed-rank test. Journal of Nonparametric Statistics 2(2), 133–153 (1993), http://www.informaworld.com/10.1080/10485259308832548

[12] van Rijsbergen, C.J.: Information Retrieval, 2nd edn. Butterworths, London (1979)

[13] Weiss, G.M.: Mining with rarity: a unifying framework. SIGKDD Explorations 6(1), 7–19 (2004), http://dblp.uni-trier.de/db/journals/sigkdd/sigkdd6.html

[14] Yang, Q., Wu, X.: 10 challenging problems in data mining research. International Journal of Information Technology and Decision Making 5(4), 597–604 (2006), http://dblp.uni-trier.de/db/journals/ijitdm/ijitdm5.html

Towards More Efficient Multi-label Classification Using Dependent and Independent Dual Space Reduction

Eakasit Pacharawongsakda and Thanaruk Theeramunkong

School of Information, Computer, and Communication Technology
Sirindhorn International Institute of Technology
Thammasat University, Thailand
{eakasit,thanaruk}@siit.tu.ac.th

Abstract. While multi-label classification can be widely applied for problems where multiple classes can be assigned to an object, its effectiveness may be sacrificed due to curse of dimensionality in the feature space and sparseness of dimensionality in the label space. Moreover, it suffers with high computational cost when there exist a high number of dimensions, as well as with lower accuracy when there are a number of noisy examples. As a solution, this paper presents two alternative methods, namely Dependent Dual Space Reduction and Independent Dual Space Reduction, to reduce dimensions in the dual spaces, i.e., the feature and label spaces, using Singular Value Decomposition (SVD). The first approach constructs the covariance matrix to represent dependency between the features and labels, project both of them into a single reduced space, and then perform prediction on the reduced space. On the other hand, the second approach handles the feature space and the label space separately by constructing a covariance matrix for each space to represent feature dependency and label dependency before performing SVD on dependency profile of each space to reduce dimension and for noise elimination and then predicting using their reduced dimensions. A number of experiments evidence that prediction on the reduced spaces for both dependent and independent reduction approaches can obtain better classification performance as well as faster computation, compared to the prediction using the original spaces. The dependent approach helps saving computational time while the independent approach tends to obtain better classification performance.

Keywords: multi-label classification, Singular Value Decomposition, SVD, dimensionality reduction, Problem Transformation

1 Introduction

In the past, most traditional classification techniques usually assumed a single category for each object to be classified by means of minimum distance. However, in some tasks it is natural to assign more than one categories to an object. For examples, some news articles can be categorized into both *politic*

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 383–394, 2012.

and *crime*, or some movies can be labeled as *action* and *comedy*, simultaneously. As a special type of task, multi-label classification was initially studied by Schapire and Singer (2000) [10] in text categorization. Later many techniques in multi-label classification have been proposed for various applications such as semantic scene classification [2], music emotion categorization [12] and automated tag recommendation [8]. However, these methods can be grouped into two main approaches: *Algorithm Adaptation* (AA) and *Problem Transformation* (PT) as suggested in [13]. The former approach modifies existing classification methods to handle multi-label data [4,10]. On the other hand, the latter approach transforms a multi-label classification task into several single classification tasks and then applies traditional classification method on each task [2,9,14].

Residing in these two main approaches, one major issue is curse of dimensionality, which causes a well-known overfitting problem. To solve this issue, many techniques have been proposed, e.g., sparse regularization [6], feature selection [17] and dimensionality reduction [15,16,18]. Among these methods, the dimensionality reduction which transforms data in a high-dimensional space to those in the lower-dimensional space, has been focused for multi-label classification problem. The dimensionality reduction in multi-label data was formerly studied by Yu et al. [16]. In their work, Multi-label Latent Semantic Indexing (MLSI) was proposed to project the original feature space into a reduced feature space. Motivated by MLSI, Multi-label Dimensionality Reduction via Dependence Maximization (MDDM) was introduced by Zhang and Zhou in [18]. In MDDM, Hilbert-Schmidt Independence Criterion (HSIC) was applied rather than LSI and its aim was to identify reduced feature space that maximizes dependency between the original feature space and the label space. Recently, Wang et al. [15] has proposed a method to extend Linear Discriminant Analysis (LDA), a well-known dimensionality reduction method, to handle multi-label data. Such methods mainly focused on how to project the original feature space into a smaller one, but still suffered with high dimensionality in the label space. By this reason, these methods usually have high time complexity in the classification task.

On the other hand, for the label space reduction, to improve the efficiency of multi-label classification, Hsu et al. [7] posed a sparseness problem that mostly occurred in the label space and then applied Compressive Sensing (CS) technique, widely used in the image processing field, to encode and decode the label space. While the encoding step of this CS method seems efficient but the decoding step does not. Toward this issue, Tai and Lin [11] proposed Principle Label Space Transformation (PLST) to transform the label space into a smaller linear label space using Singular Value Decomposition (SVD) [5] with a simple threshold setting (i.e., 0.5). More recently, Bi and Kwok (2011) [1] extended the PLST to handle the labels which are organized in the form of a tree or directed acyclic graph (DAG). Although the PLST-based methods are effective by reducing dimensions in the label space, it seems not handle neither the curse of dimensionality in the feature space nor the correlation (dependency) among labels in the label space.

Toward these issues, this paper presents an approach that considers both the curse of dimensionality problem in the feature space and the sparseness problem in the label space. Moreover, the dependency profile among features and labels, the dependency profile among features and the dependency profile among labels are also taken into account. Two alternative methods, namely Dependent Dual Space Reduction (DDSR) and Independent Dual Space Reduction (IDSR), are proposed to reduce dimensions in the dual spaces to eliminate redundancy as well as noise using Singular Value Decomposition (SVD) for better prediction and lower computational cost.

In the rest of this paper, Section 2 gives a formal description of the multi-label classification task and literature review to the SVD method. Section 3 presents two dual dimensionality reduction approaches, Dependent Dual Space Reduction (DDSR) and Independent Dual Space Reduction (IDSR). The multi-label benchmark datasets and experimental settings are described in Section 4. In Section 5, the experimental results using seven datasets are given and finally Section 6 provides conclusion of this work.

2 Preliminaries

2.1 Definition of Multi-label Classification Task

Let $\mathcal{X} = \mathbb{R}^M$ and $\mathcal{Y} = \{0,1\}^L$ be an M-dimensional feature space and L-dimensional binary label space, where M is the number of features and L is a number of possible labels, i.e. classes. Let $\mathcal{D} = \{\langle \mathbf{x}_1, \mathbf{y}_1 \rangle, \langle \mathbf{x}_2, \mathbf{y}_2 \rangle, ..., \langle \mathbf{x}_N, \mathbf{y}_N \rangle\}$ is a set of N objects (e.g., documents, images, etc.) in a training dataset, where $\mathbf{x}_i \in \mathcal{X}$ is a feature vector that represents an i-th object and $\mathbf{y}_i \in \mathcal{Y}$ is a label vector with the length of L, $[y_{i1}, y_{i2}, ..., y_{iL}]$. Here, y_{ij} indicates whether the i-th object belongs (1) or not (0) to the j-th class (j-th label or not).

In general, two main phases are exploited in a multi-label classification problem: (1) *model training* phase and (2) *classification* phase. The goal of the *model training* phase is to build a classification model that can predict the label vector $\mathbf{y}_t$ for a new object with the feature vector $\mathbf{x}_t$. This classification model is a mapping function $\mathcal{H} : \mathbb{R}^M \rightarrow \{0,1\}^L$ can predict a target value closest to its actual value in total. The *classification* phase uses this classification model to assign labels. For convenience, $\mathbf{X}_{N \times M} = [\mathbf{x}_1, ..., \mathbf{x}_N]^T$ denotes the *feature matrix* with N rows and M columns and $\mathbf{Y}_{N \times L} = [\mathbf{y}_1, ..., \mathbf{y}_N]^T$ represents the *label matrix* with N rows and L columns , where $[\cdot]^T$ denotes matrix transpose.

2.2 Singular Value Decomposition (SVD)

This subsection gives a brief introduction to SVD, which was developed as a method for dimensionality reduction using a least-squared technique [5]. The SVD transforms a feature matrix $\mathbf{X}$ to a lower-dimensional matrix $\mathbf{X}'$ such that the distance between the original matrix and a matrix in a lower-dimensional space (i.e., the 2-norm $\| \mathbf{X} - \mathbf{X}' \|_2$) are minimum.

Generally, a feature matrix $\mathbf{X}$ can be decomposed into the product of three matrices as shown in (1).

$$\mathbf{X}_{N\times M} = \mathbf{U}_{N\times M} \times \mathbf{\Sigma}_{M\times M} \times \mathbf{V}^T_{M\times M}, \tag{1}$$

where N is a number of objects, M is a number of features and $M < N$. The matrices $\mathbf{U}$ and $\mathbf{V}$ are two orthogonal matrices, where $\mathbf{U}^T \times \mathbf{U} = \mathbf{I}$ and $\mathbf{V}^T \times \mathbf{V} = \mathbf{I}$. The columns in the matrix $\mathbf{U}$ are called the *left singular vectors* while as the columns in matrix $\mathbf{V}$ are called the *right singular vectors*. The matrix $\mathbf{\Sigma}$ is a diagonal matrix, where $\mathbf{\Sigma}_{i,j} = 0$ for $i \neq j$, and the diagonal elements of $\mathbf{\Sigma}$ are the singular values of matrix $\mathbf{X}$. The singular values in the matrix $\mathbf{\Sigma}$ are sorted by descending order such that $\mathbf{\Sigma}_{1,1} \geq \mathbf{\Sigma}_{2,2} \geq ... \geq \mathbf{\Sigma}_{M,M}$. To discard noise, it is possible to ignore singular values less than $\mathbf{\Sigma}_{K,K}$, where $K \ll M$. By this ignorance the three matrices are reduced to (2).

$$\mathbf{X}'_{N\times M} = \mathbf{U}'_{N\times K} \times \mathbf{\Sigma}'_{K\times K} \times \mathbf{V}'^T_{M\times K}, \tag{2}$$

where $\mathbf{X}'_{N\times M}$ is expected to be close $\mathbf{X}_{N\times M}$, i.e. $\| \mathbf{X} - \mathbf{X}' \|_2 < \delta$, $\mathbf{U}'_{N\times K}$ is a reduced matrix of $\mathbf{U}_{N\times M}$, $\mathbf{\Sigma}'_{K\times K}$ is the reduced version of $\mathbf{\Sigma}_{M\times M}$ from M to K dimensions and $\mathbf{V}'_{M\times K}$ is a reduced matrix of $\mathbf{V}_{M\times M}$.

In the next section, we show our two approaches that deploy the SVD technique to construct lower-dimensional space for both features and labels for multi-label classification.

3 Two Proposed Approaches

As mentioned earlier, most of previous approaches were presented to handle either the problem of a curse of dimensionality in the feature space or the sparseness problem in the label space.

This work presents two alternative approaches to deal with these aforementioned problems. In both approaches, the Singular Value Decomposition (SVD) is used to project both feature and label spaces into reduced spaces then a classification method can be applied. In the *classification* phase, on the other hand, SVD is used to reconstruct the original higher-dimensional label space from the prediction result in the constructed lower-dimensional label space.

In this work, we propose two alternative methods, called Dependent Dual Space Reduction (DDSR) and Independent Dual Space Reduction (IDSR). To promote the dependency among features and labels, DDSR computes dependency matrix between features and labels then applies SVD to eliminate the less correlated data. These lower-dimensional matrices computed from SVD are used to project both feature and label spaces into a common lower-dimensional space. While the DDSR approach retains only data with high correlated between features and labels, it neither considers dependency among features nor dependency among labels. As our second proposed method, the Independent Dual Space Reduction (IDSR) uses the feature dependency matrix built from the feature space and label dependency matrix computed from the label space as two projection

matrices. After that two independent SVDs are applied to these matrices and project feature space and label space to lower-dimensional spaces. The prediction can be done on lower-dimensional spaces before transforming back to the original space. The next subsections describe DDSR and IDSR in order.

3.1 Dependent Dual Space Reduction (DDSR)

As previously mentioned, it is possible to utilize the characteristic that the feature space and the label space may have some dependency with each other. While it is possible to characterize a dependency among feature and label spaces, for example, cosine similarity, entropy and symmetric uncertainty, with limited space, we considered only covariance matrix. In this work, both spaces can be simultaneously compressed by performing SVD on the feature-label covariance, viewed as a dependency profile between features and labels. Equation (3) shows construction of a covariance matrix $\mathbf{S}_{M\times L}$ to represent a dependency between feature and label spaces.

$$\mathbf{S}_{M\times L} = E[(\mathbf{X}_{N\times M} - E[\mathbf{X}_{N\times M}])^T(\mathbf{Y}_{N\times L} - E[\mathbf{Y}_{N\times L}])], \tag{3}$$

where $\mathbf{X}$ is the feature matrix, $\mathbf{Y}$ is the label matrix and $E[\cdot]$ is an expected value of the matrix.

Applying SVD, the covariance matrix $\mathbf{S}_{M\times L}$ is later decomposed to matrices $\mathbf{U}$, $\mathbf{\Sigma}$ and $\mathbf{V}$. To retain only significant dimensions and reduce noise, the first K ($\leq min(M, L)$) dimensions from matrices $\mathbf{U}$ and $\mathbf{V}$ are selected as $\mathbf{U'}_{M\times K}$ and $\mathbf{V'}_{M\times K}$. Here, a lower-dimensional feature matrix $\mathbf{X'}$, can be created as $\mathbf{X'}_{N\times K} = \mathbf{X}_{N\times M} \times \mathbf{U'}_{M\times K}$. In the same way, a lower-dimensional label matrix $\mathbf{Y'}$ can be computed as $\mathbf{Y'}_{N\times K} = \mathbf{Y}_{N\times L} \times \mathbf{V'}_{L\times K}$. These tasks constitute the *pre-processing* phase.

In the next step, these two lower-dimensional matrices, $\mathbf{X'}$ and $\mathbf{Y'}$, are used for building a classification model. Among existing methods on multi-label classification, Binary Relevance (BR) is a simple approach and widely used. BR simply reduces the multi-label classification task to a set of binary classifications and then builds a classification model for each class. However, in this approach, the projected label matrix $\mathbf{Y'}$ contains numeric values rather than discrete classes. By this situation, a regression method can be applied to estimate these numeric values. While the projected label matrix $\mathbf{Y'}$ has K dimensions, it is possible to construct a regression model for each dimension. That is, K regression models are constructed for K lower-dimensions. Moreover, each model, later denoted by $r_k(\mathbf{X'})$ is a regression model built for predicting each column $\mathbf{Y'}[k]$ using the matrix $\mathbf{X'}$. While the regression model returns continuous values, we propose a method to find the optimal threshold for mapping a continuous value to binary decision (0 or 1) as described in Section 3.3.

In the *classification* phase, firstly a test feature vector $\hat{\mathbf{X}}$ is transformed to the lower-dimensional feature vector $\hat{\mathbf{X}}'$ using $\hat{\mathbf{X}}'_{1\times K} = \hat{\mathbf{X}}_{1\times M} \times \mathbf{U'}_{M\times K}$. Then this vector is fed to a series of regression models $r(\hat{\mathbf{X}}')$ to estimate the numeric value in each dimension of the predicted lower-dimensional label vector $\hat{\mathbf{Y}}'_{1\times K}[k]$.

After that, a matrix $\mathbf{V'}^T$, an orthogonal matrix of the matrix $\mathbf{V'}$, is multiplied to reconstruct the lower-dimensional label vector $\hat{\mathbf{Y}}_{1\times K}$ back to the higher-dimensional label vector $\hat{\mathbf{Y}}_{1\times L}$. Next the predicted values in the label vector $\hat{\mathbf{Y}}_{1\times L}$ are rounded to the value in {0,1} by the predefined threshold. The set of predicted multiple labels is the union of the dimensions which have the value of 1.

3.2 Independent Dual Space Reduction (IDSR)

As opposed to the former approach, the Independent Dual Space Reduction (IDSR) approach presents how to use two independent SVDs for transforming the feature space and label space into the two lower-dimensional spaces similar to DDSR even there are several possibilities of dependency calculation. In this work, to consider the dependency in the feature space, the covariance matrix $\mathbf{S}_{M\times M}$ is computed from the feature matrix $\mathbf{X}_{N\times M}$. On the other hand, the dependency among labels in the label space can be derived by calculating the covariance matrix $\mathbf{R}_{L\times L}$.

In the *pre-processing* phase of this approach, a feature dependency matrix $\mathbf{S}_{M\times M}$ is built from a feature matrix $\mathbf{X}_{N\times M}$ and then it is decomposed to three matrices $\mathbf{U_x}$, $\mathbf{\Sigma_x}$ and $\mathbf{V_x}$ and select the top D dimensions from the matrix $\mathbf{U_x}$. Then the lower-dimensional feature matrix $\mathbf{X'}$ can be constructed by $\mathbf{X'}_{N\times D} = \mathbf{X}_{N\times M} \times \mathbf{U'}_{\mathbf{x}M\times D}$. The label dependency matrix $\mathbf{R}_{L\times L}$ is constructed from the label matrix $\mathbf{Y}$. Likewise, this matrix is decomposed to three matrices $\mathbf{U_y}$, $\mathbf{\Sigma_y}$ and $\mathbf{V_y}$ and the top K dimensions are selected from the matrix $\mathbf{U_y}$. The lower-dimensional label matrix $\mathbf{Y'}$ can be formulated by $\mathbf{Y'}_{N\times K} = \mathbf{Y}_{N\times L} \times \mathbf{U'}_{\mathbf{y}L\times K}$. While the original label matrix $\mathbf{Y}_{N\times L}$ contains either 0 or 1 as its members, its lower-dimensional label matrix $\mathbf{Y'}_{N\times K}$ may include non-binary numeric values. Moreover, it is not necessary that the dimension of the lower-dimensional feature space D and that of the lower-dimensional label space K are identical. Note that this condition is not the same with the DDSR approach, where D always equals to K. After that, as the *model training* phase, we can construct K regression models to predict $\mathbf{Y'}_{N\times K}$ from $\mathbf{X'}_{N\times D}$. Note that each regression model is for each of K dimensions of $\mathbf{Y'}$. To transform a numeric value to a binary value a threshold is established. Section 3.3 describes our proposed method for searching the best threshold for each label. This step is done in the *model training* phase.

In the *classification* phase, the feature vector $\hat{\mathbf{X}}$ of an unseen object will be reduced to the lower-dimensional feature vector $\hat{\mathbf{X}}'$ using $\hat{\mathbf{X}}'_{1\times D} = \hat{\mathbf{X}}_{1\times M} \times \mathbf{U'}_{\mathbf{x}M\times D}$. Then the regression models estimate the numeric value in the lower-dimensional label vector $\hat{\mathbf{Y}}'$ based on the feature vector $\hat{\mathbf{X}}'$. Next the matrix $\mathbf{U'_y}^T$, an orthogonal matrix of the matrix $\mathbf{U'_y}$, is used to reconstruct the original higher-dimensional label vector $\hat{\mathbf{Y}}$ from the prediction result in the lower-dimensional label vector $\hat{\mathbf{Y}}'$ i.e., $\hat{\mathbf{Y}}_{N\times L} = \hat{\mathbf{Y}}_{N\times K} \times \mathbf{U'}^T_{\mathbf{y}L\times K}$. To assign labels to an unseen object, the prediction values in the label vector $\hat{\mathbf{Y}}$ need to be rounded to {0,1}. At this point, the threshold found in the *model training* phase can be applied. Finally, the assigned label set is the union set of the dimensions that have the value of 1.

3.3 Threshold Selection

As stated above, by the orthogonal property of SVD, it can be used to reconstruct an original label space from a lower-dimensional label space. As the result, the reconstructed label vector may include non-binary values. To interpret the values as binary decision, a threshold need to be set to map these values to either 0 or 1 for representing whether the object belongs to the class or not. As a naive approach, the fixed value of 0.5 is used to assign 0 if the value is less than 0.5, otherwise 1 [11]. As a more efficient method, it is possible to apply an adaptive threshold. In this work, we propose a method to determine an optimal threshold by selecting the value that maximizes classification accuracy in the training dataset that is similar to the mechanism in Han et al [6]. In other words, the threshold selection is done by first sorting prediction values in each label dimension in a descending order and examine performance (e.g., *macro F-measure*) for each rank position from the top to the bottom to find the point that maximize the performance. Then, the threshold for binary decision is set based on that point.

4 Datasets and Experimental Settings

To evaluate the performance of our two proposed approaches, the benchmark multi-label datasets are downloaded from MULAN[1]. Table 1 shows the characteristics of seven multi-label datasets. For each dataset, N, M and L denote the total number of objects, the number of features and the number of labels, respectively. L_C represents the *label cardinality*, the average number of labels per example and L_D stands for *label density*, the normalized value of *label cardinality* as introduced by Read et al [9].

Table 1. Characteristics of the datasets used in our experiments

Dataset	Domain	N	M		L	L_C	L_D
			Nominal	Numeric			
bibtex	text	7,395	1,836	-	159	2.402	0.015
corel5k	images	5,000	499	-	374	3.522	0.009
enron	text	1,702	1,001	-	53	3.378	0.064
medical	text	978	1,449	-	45	1.245	0.028
emotions	music	593	-	72	6	1.869	0.311
scene	image	2,407	-	294	6	1.074	0.179
yeast	biology	2,417	-	103	14	4.237	0.303

Since each object in the dataset can be associated with multiple labels simultaneously, the traditional evaluation metric of single-label classification could not be applied. The well-known multi-label evaluation metrics are of two types [9]. As the first type, a *label-based* metric evaluates each label separately such

[1] http://mulan.sourceforge.net/datasets.html

as *hamming loss* and *macro F-measure*. As the second type, a *label set-based* metric considers a set of labels simultaneously, i.e., *accuracy* and *0/1 loss*. In this work, *hamming loss*, *macro F-measure*, *accuracy* and *0/1 loss* are used to assess the effectiveness of the multi-label classification methods. Their detailed descriptions can be found in several literatures such as those in Read et al [9].

Table 2. Performance comparison (mean) between BR, BR+, CC, PLST, DDSR, and IDSR in terms of *hamming loss (HL), macro F-measure (F1), accuracy, 0/1 loss* on the seven datasets. (↓ indicates the smaller the better; ↑ indicates the larger the better; † denotes the nominal-feature dataset and ‡ denotes the numeric-feature dataset, the superscript (x,y) shows the percentage of dimensions reduced from the original ones for the feature space and that for the label space.)

Dataset	Metrics	BR	BR+	CC	PLST	DDSR	IDSR
bibtex†	HL ↓	0.0172	0.0163	0.0166	$0.0155^{[20]}$	$\mathbf{0.0137}^{(9,100)}$	$0.0140^{(40,100)}$
	F1 ↑	0.0047	0.0022	0.0038	$0.0075^{[100]}$	$0.3949^{(9,100)}$	$\mathbf{0.4025}^{(80,100)}$
	Accuracy ↑	0.0000	0.0001	0.0001	$0.0012^{[100]}$	$0.1543^{(3,40)}$	$\mathbf{0.3699}^{(80,100)}$
	0/1 Loss ↓	1.0000	1.0000	1.0000	$0.9999^{[40]}$	$\mathbf{0.8270}^{(9,100)}$	$0.8301^{(60,100)}$
corel5k†	HL ↓	0.0094	0.0195	0.0094	$\mathbf{0.0094}^{[20]}$	$0.0145^{(30,40)}$	$0.0150^{(60,100)}$
	F1 ↑	0.0284	0.0710	0.0354	$0.0286^{[40]}$	$\mathbf{0.1215}^{(15,20)}$	$0.1143^{(60,100)}$
	Accuracy ↑	0.0528	0.1432	0.0584	$0.0533^{[40]}$	$0.1479^{(60,80)}$	$\mathbf{0.1502}^{(60,100)}$
	0/1 Loss ↓	0.9948	0.9938	**0.9918**	$0.9944^{[20]}$	$0.9944^{(45,60)}$	$0.9948^{(60,100)}$
enron†	HL ↓	0.1019	0.0997	0.1014	$0.0721^{[20]}$	$\mathbf{0.0529}^{(3,60)}$	$0.0532^{(20,100)}$
	F1 ↑	0.1957	0.1882	0.1920	$0.2170^{[40]}$	$\mathbf{0.3118}^{(5,100)}$	$0.2926^{(20,100)}$
	Accuracy ↑	0.3328	0.3274	0.3317	$0.3475^{[20]}$	$\mathbf{0.4723}^{(5,100)}$	$0.4617^{(20,100)}$
	0/1 Loss ↓	0.9089	0.9083	0.9066	$0.9077^{[60]}$	$\mathbf{0.8713}^{(5,100)}$	$0.8766^{(60,100)}$
medical†	HL ↓	0.0996	0.0943	0.0974	$0.0556^{[20]}$	$0.0114^{(3,100)}$	$\mathbf{0.0107}^{(20,100)}$
	F1 ↑	0.3383	0.3370	0.3351	$0.3411^{[60]}$	$0.6485^{(3,100)}$	$\mathbf{0.6815}^{(40,100)}$
	Accuracy ↑	0.4017	0.3997	0.4060	$0.4074^{[40]}$	$0.7288^{(3,100)}$	$\mathbf{0.7603}^{(20,100)}$
	0/1 Loss ↓	0.7167	0.7085	0.7095	$0.7167^{[100]}$	$0.3731^{(3,100)}$	$\mathbf{0.3507}^{(20,100)}$
emotions‡	HL ↓	0.2068	0.2264	0.2211	$\mathbf{0.2037}^{[60]}$	$0.2728^{(8,100)}$	$0.2152^{(100,60)}$
	F1 ↑	0.6317	0.6673	0.6243	$0.6339^{[60]}$	$0.6455^{(8,100)}$	$\mathbf{0.6804}^{(100,20)}$
	Accuracy ↑	0.5091	**0.5537**	0.5176	$0.5091^{[100]}$	$0.5090^{(8,100)}$	$0.5534^{(60,40)}$
	0/1 Loss ↓	0.7467	**0.7267**	0.7451	$0.7300^{[60]}$	$0.8293^{(8,100)}$	$0.7417^{(60,60)}$
scene‡	HL ↓	**0.1105**	0.2583	0.1162	$\mathbf{0.1105}^{[100]}$	$0.1190^{(2,80)}$	$0.1113^{(60,80)}$
	F1 ↑	0.6480	0.5351	0.6903	$0.6480^{[100]}$	$0.7046^{(2,100)}$	$\mathbf{0.7201}^{(20,80)}$
	Accuracy ↑	0.5302	0.5744	**0.6579**	$0.5302^{[100]}$	$0.6109^{(2,80)}$	$0.6451^{(20,80)}$
	0/1 Loss ↓	0.5186	0.5148	**0.3831**	$0.5186^{[100]}$	$0.5106^{(2,80)}$	$0.4778^{(40,80)}$
yeast‡	HL ↓	**0.2008**	0.2229	0.2165	$0.2491^{[20]}$	$0.2807^{(14,100)}$	$0.2626^{(80,100)}$
	F1 ↑	0.4455	0.4053	0.4404	$0.2688^{[20]}$	$0.4926^{(3,20)}$	$\mathbf{0.4927}^{(40,100)}$
	Accuracy ↑	0.5019	0.4838	0.4796	$0.3425^{[20]}$	$0.4884^{(8,60)}$	$\mathbf{0.5021}^{(60,60)}$
	0/1 Loss ↓	0.8510	0.8564	**0.8105**	$0.9785^{[40]}$	$0.9259^{(14,100)}$	$0.9086^{(80,100)}$

In this work, DDSR and IDSR are compared with four multi-label classification techniques; BR, BR+ [3], CC [9] and PLST [11]. BR+ (Binary Relevance with label dependency consideration) and CC (Classifier Chains) are two well-known methods, which incorporate label dependency in multi-label classification. PLST (Principle Label Space Transformation) is an efficient algorithm that uses the reduction of label space dimension. Using ten-fold cross validation method, the results of the four evaluation metrics and the execution time are recorded and shown in Table 2 and 3, respectively. All multi-label classification methods used in this work is implemented in R environment version 2.11.1[2] and linear

[2] http://www.R-project.org/

regression is used for the regression model in the *model training* phase. For PLST and IDSR method, we experiments with K ranging from 20% to 100% of the dimension of the original label matrix, with 20% as interval. Likewise, the parameter D is also varied from 20% to 100% of the dimension of the original feature matrix, with 20% as interval. Though, the K parameter in DDSR approach is calculated from the minimum value between a number of features and labels, this parameter is also used the same criteria as PLST and IDSR method. To compare the computational time, all methods were performed on the AMD Opteron Quad Core 8356 1.1 GHz Processor with 512 KB of cache, 64GB RAM.

5 Experimental Results

To evaluate our proposed approaches, seven datasets are used to compare performance of BR, BR+, CC, PLST, DDSR and IDSR. Table 2 reports the best value for each evaluation metric computed from all datasets. The numbers in the superscript (x,y) represents the percentages of dimensions reduced in the feature space and that in the label space, respectively. In the DDSR method, the maximum number of reduced dimensions K cannot excess the minimum between the number of features (M) and the number of labels (L) since the reduced dimension has the same size of both feature and label space. The superscript [y] in the PLST approach means the percentage of reduce labels, compared to the original. Note that PLST does not reduce the feature space. In the Table 2, the best value for each row is emphasized by bold font.

From the table, we can make some observations as follows. First, we observe that both DDSR and IDSR give comparable performance in terms of *hamming loss*, compared to BR, BR+, CC and PLST. On the other hand, DDSR and IDSR approaches gain an average gap of 16% *macro F-measure* increment. Moreover, the DDSR approach shows an average gap of 11% *accuracy* improvement while the IDSR approach improves with an average gap of 15%. Likewise, the DDSR approach can reduce the *0/1 loss* with decrement of 5% while IDSR can reduce with 8% gap. Note that the *medical* dataset whose number of features are greater than the number of objects, gives the maximum improvement when both spaces are reduced.

As shown in Table 3, the execution time of the DDSR approach was reduced with a factor of 10, compared to PLST and approximately 100 times, compared to the traditional BR approach. Likewise, the IDSR approach with lower-dimensional features used less time than the PLST and the BR method. We can conclude that our two proposed methods, DDSR and IDSR, could transform the feature and label spaces into the reduced spaces with less computational time than the traditional BR, BR+, CC and PLST. As an additional observation, IDSR is better than DDSR in several datasets while DDSR can be executed faster than IDSR. The smaller K and D are, the faster we can compute.

In more details, Table 4 presents the complexity of learning process. However, the time used for the transformation process and covariance calculation is trivial. The N, M and L denote the number of objects, features and labels, respectively.

Table 3. Average execution time (in seconds) on the seven datasets. † denotes the nominal-feature dataset and ‡ denotes the numeric-feature dataset. Here, D is the reduced number of feature dimensions. K is the reduced number of label dimensions.

Dataset	Method	K		
		20%×L	60%×L	100%×L
bibtex†	BR	-	-	40442.05
	BR+	-	-	33537.76
	CC	-	-	19700.05
	PLST	7101.01	21961.42	29815.07
	DDSR	69.85	126.56	283.36
	IDSR (D=20%×M)	1338.02	2360.49	3855.54
	IDSR (D=60%×M)	4656.35	14636.93	18787.84
	IDSR (D=100%×M)	14800.45	27392.90	46642.77
corel5k†	BR	-	-	5721.81
	BR+	-	-	11971.31
	CC	-	-	7007.41
	PLST	1172.75	3174.11	5386.21
	DDSR	69.73	366.07	1265.54
	IDSR (D=20%×M)	364.17	560.04	818.53
	IDSR (D=60%×M)	817.59	2667.54	4572.47
	IDSR (D=100%×M)	1905.29	5273.13	6397.23
enron†	BR	-	-	656.94
	BR+	-	-	1904.89
	CC	-	-	799.58
	PLST	197.43	537.04	824.20
	DDSR	4.62	6.57	11.56
	IDSR (D=20%×M)	29.85	41.03	50.12
	IDSR (D=60%×M)	60.48	120.46	175.46
	IDSR (D=100%×M)	99.27	233.00	355.83
medical†	BR	-	-	1353.93
	BR+	-	-	2896.23
	CC	-	-	1414.56
	PLST	192.49	509.45	1240.37
	DDSR	1.35	2.57	4.76
	IDSR (D=20%×M)	238.56	256.15	234.38
	IDSR (D=60%×M)	170.31	324.86	383.25
	IDSR (D=100%×M)	449.11	735.24	982.16
emotions‡	BR	-	-	1.10
	BR+	-	-	4.09
	CC	-	-	0.87
	PLST	0.19	0.47	0.79
	DDSR	0.41	0.47	0.53
	IDSR (D=20%×M)	0.48	0.56	0.59
	IDSR (D=60%×M)	0.51	0.68	0.85
	IDSR (D=100%×M)	0.57	1.00	1.27
scene‡	BR	-	-	9.13
	BR+	-	-	57.33
	CC	-	-	22.26
	PLST	2.31	12.08	10.80
	DDSR	2.70	2.79	2.87
	IDSR (D=20%×M)	3.52	4.09	4.47
	IDSR (D=60%×M)	4.80	6.84	8.13
	IDSR (D=100%×M)	6.54	10.72	13.24
yeast‡	BR	-	-	16.23
	BR+	-	-	37.98
	CC	-	-	15.62
	PLST	3.93	9.23	10.15
	DDSR	10.52	10.82	11.37
	IDSR (D=20%×M)	11.68	11.99	12.04
	IDSR (D=60%×M)	11.82	12.85	14.26
	IDSR (D=100%×M)	12.33	14.15	16.20

For our two approaches, D and K are the reduced number of dimensions. The $f(X, Y)$ is the complexity of the model that depends on the number of objects (X) and the number of features (Y). When linear regression is applied for the *model training* phase, it requires $O(4XY^2 + X^3 + 2XY)$ and $O(Y)$ for the *classification* phase. We can observe that the BR+ method is recognized as the slowest algorithm since it appends labels to the feature space for incorporating label dependency and it requires two learning process, initial prediction step and final prediction step, to complete the classification process. On the other hand,

Table 4. The learning complexity of BR, BR+, CC, PLST, DDSR and IDSR. Note that the complexity is in the function $f(X, Y)$, where X is the number of objects and Y is the number of features.

Methods	Complexity (O)
BR	$O(L \times f(N, M))$
BR+	$O((L \times f(N, M)) + (L \times f(N, (M + L - 1))))$
CC	$O(L \times f(N, (M + L/2)))$
PLST	$O(K \times f(N, M))$
DDSR	$O(K \times f(N, K))$
IDSR	$O(K \times f(N, D))$

our DDSR approach is the fastest method because the feature and label spaces are transformed to the lower-dimensional space before classification technique is applied.

6 Conclusion

This paper presents two alternative approaches to handle the curse of dimensionality problem in the feature space as well as the sparseness problem in the label space. The Dependent Dual Dimensionality Reduction (DDSR) considers the dependency between feature and label spaces before transforming the feature and label spaces into a single reduced space. On the other hand, the Independent Dual Space Reduction (IDSR) approach transforms the feature space and label space into the two lower-dimensionality spaces. Experiments with a broad range of multi-label datasets show that our two proposed approaches achieve a better performance, compared to PLST and BR, as well as other recent methods such as Classifier Chains (CC) and BRplus (BR+). In addition, the DDSR approach helps saving computational time while the IDSR approach tends to obtain better classification performance As our future work, we will analyze three dependencies, feature-label, feature-feature, and label-label, in detail. The ensemble of these dependencies may help in improving the performance.

Acknowledgement. This work has been supported by the TRF Royal Golden Jubilee Ph.D. Program [PHD/0304/2551] and partially supported by the National Research University Project of Thailand Office of Higher Education Commission as well as the National Electronics and Computer Technology Center (NECTEC) under Project Number NT-B-22-KE-38-54-01.

References

1. Bi, W., Kwok, J.: Multi-label classification on tree- and dag-structured hierarchies. In: Getoor, L., Scheffer, T. (eds.) Proceedings of the 28th International Conference on Machine Learning (ICML 2011), pp. 17–24. ACM, New York (2011)
2. Boutell, M., Luo, J., Shen, X., Brown, C.: Learning multi-label scene classification. Pattern Recognition 37(9), 1757–1771 (2004)

3. Cherman, E.A., Metz, J., Monard, M.C.: Incorporating label dependency into the binary relevance framework for multi-label classification. Expert Systems with Applications (2011)
4. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Proceedings of the Advances in Neural Information Processing Systems, vol. 14, pp. 681–687. MIT Press (2001)
5. Golub, G., Reinsch, C.: Singular value decomposition and least squares solutions. Numerische Mathematik 14, 403–420 (1970)
6. Han, Y., Wu, F., Jia, J., Zhuang, Y., Yu, B.: Multi-task sparse discriminant analysis (mtsda) with overlapping categories. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, pp. 469–474 (2010)
7. Hsu, D., Kakade, S., Langford, J., Zhang, T.: Multi-label prediction via compressed sensing. In: Proceedings of the Advances in Neural Information Processing Systems, vol. 22, pp. 772–780 (2009)
8. Katakis, I., Tsoumakas, G., Vlahavas, I.: Multilabel text classification for automated tag suggestion. In: Proceedings of the the ECML/PKDD 2008 Discovery Challenge (2008)
9. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine Learning, 1–27 (2011)
10. Schapire, R., Singer, Y.: Boostexter: A boosting-based system for text categorization. Machine Learning 39(2/3), 135–168 (2000)
11. Tai, F., Lin, H.T.: Multi-label classification with principle label space transformation. In: Proceedings of the 2nd International Workshop on Learning from Multi-Label Data (MLD 2010), pp. 45–52 (2010)
12. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multi-label classification of music into emotions. In: Proceedings of the International Symposium/Conference on Music Information Retrieval, pp. 325–330 (2008)
13. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining Multi-label Data. In: Data Mining and Knowledge Discovery Handbook, 2nd edn. Springer (2010)
14. Tsoumakas, G., Katakis, I., Vlahavas, I.: Random k-labelsets for multilabel classification. IEEE Transactions on Knowledge and Data Engineering 23, 1079–1089 (2011)
15. Wang, H., Ding, C., Huang, H.: Multi-label Linear Discriminant Analysis. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 126–139. Springer, Heidelberg (2010)
16. Yu, K., Yu, S., Tresp, V.: Multi-label informed latent semantic indexing. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 258–265 (2005)
17. Zhang, M.L., Pea, J.M., Robles, V.: Feature selection for multi-label naive bayes classification. Information Science, 3218–3229 (2009)
18. Zhang, Y., Zhou, Z.H.: Multilabel dimensionality reduction via dependence maximization. ACM Transactions on Knowledge Discovery from Data (TKDD) 4(3), 1–21 (2010)

Automatic Identification of Protagonist in Fairy Tales Using Verb

Hui-Ngo Goh, Lay-Ki Soon, and Su-Cheng Haw

Faculty of Computing and Informatics, Multimedia University, Jalan Multimedia, 63100 Cyberjaya Selangor, Malaysia
{hngoh,lksoon,sucheng}@mmu.edu.my

Abstract. Named entity recognition (NER) has been a well-studied problem in the area of text mining for locating atomic element into predefined categories, where "name of people" is one of the most commonly studied categories. Numerous new NER techniques have been unfolded to accommodate the needs of the application developed. However, most research works carried out focused on non-fiction domain. Fiction domain exhibits complexity and uncertainty in locating protagonist as it represents name of person in a diverse spectrums, ranging from living things (animals, plants, person) to non-living things (vehicle, furniture). This paper proposes automated protagonist identification in fiction domain, particularly in fairy tales. Verb has been used as a determinant in substantiating the existence of protagonist with the assistance of WordNet. The experimental results show that it is viable to use verb in identifying named entity, particularly "people" category and it can be applied in a small text size environment.

Keywords: Named entity recognition, characters, fairy tales, text mining.

1 Introduction and Motivation

Named entity recognition (NER) is a well-studied research in the area of information extraction (IE) aiming to locate and extract significant atomic elements in text into predefined categories. The most common studied categories are "name of people, organization and location" [1], [2] and [3], "date, time and phone" [4], "name of person, diploma, organization and research" [5] and many more entities as of interest of the application intended to be built.

Andrew e*t al.* used list of features, lexicon (augment using web search engine) and conditional random fields (CRF), a machine learning probabilistic approach to extract named entities in structured texts of CoNLL03 (name of person, location, organization and miscellaneous) [2]. Einat *et al.* focused solely in extracting personal names from informal text (email) using CRF and dictionary to enhance the names extraction [6]. The performance results vary among the chosen email corpora due to the free writing style in informal text and insufficient training data to produce good model for NER. Satoshi *et al.* manually hand-crafted about 1400 rules and 130,000 instances of dictionary to extract 200 categories of named entity covering generally Japanese newspaper domain [7]. In 2010, Laura *et al.* proposed domain

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 395–406, 2012.

adaptation of rule-based annotator to enhance domain customization for NER, a domain-independent CoreNER library of 104 features definition were being crafted manually to tailor different application domains need [1]. Public datasets of CoNLL03, Enron and ACE05 were used to train and test the "person, location and organization" entities. However, it is still manual and time consuming.

Character level model [3] used Hidden Markov Model (HMM) and Maximum-entropy Conditional Markov Model to inspect each letter in identifying named entity in ConLL, the character emission model is based on the *n*-gram proper-name classification engine [8] and state transition chaining is used to identify named entity boundary and classify them into predefined categories. Le *et al.* studied the use of inductive logic programming to extract named entities (name, diploma, organization, research) in Vietnamese language [5]. 80 Vietnamese homepages of scientist that were tagged manually and a set of features were used to train to generate a set of extraction rules to extract named entities in chosen test corpus. Javier *et al.* discussed the impact of coverage, reliability and independent number of features in extracting name of person [9]. Machine-learning algorithm, NER and classification were used to studied the mentioned impact, in the case of NER, combination of Stanford NE Recogniser (machine learning) and OAK (rule based English analyzer) were used to detect NE. It generates good performance results if all NE features are being used in the training process when producing trained model for NE recognition. Michael *et al.* studied further details of breaking down "name of person" into sub-categories such as "politician" and "entertainer", topic signatures and Wordnet are used to enhance the trained model using supervised machine learning [10].

All the above mentioned NER techniques are mainly constrained by two major issues as discussed below:

(1) *Recognition approach*: The evolutionary of NER begins with manual effort to semi-automatic, and then to automatic approaches. Manual NER requires excessive amount of time and resources from domain expert and knowledge engineer to hand-coded the rules manually. In such approach, existing text mining resources such as WordNet and dictionaries are always in used to speed up the manual NER process [7]. Knowing that manual construction of NER rules exhibits a promising performance results due to the intentionality of recognizing NE in the domain studied, Laura *et al.* explored the domain customization using rule-based annotator; a set of universal rules is needed to accommodate investigated domain needs [1]. However, flexibility and scalability is still the main issue to be dissolved in manual NER. Semi-automatic NER begins with seed (manual selection) NE. Often, machine learning is used to train the seed NE to generate a model for NER. The quality of chosen seed data will greatly impact the trained model for NE recognition. Automatic NER implies fully recognition without human intervention. It is not an easy task as each domain exhibits differently in term of context and structural text representation.

(2) *Nature of the domain*: Research domain done in the area of NER can be classified into fiction based and non-fiction based. Fiction implies literary work which is based on imagination and not necessary on facts, e.g. novel and fairy tales, whereas non-fiction denotes representation of a subject which is presented as fact, such as

manual, news-wired and tourism website. Most NERs developed are in non-fiction based. Non-fiction based exhibits certain patterns in identifying NE, for instance, name of person may start with designator, capital letter of the first character, naming in a human way. However, fiction based exhibits complexity and uncertainty in locating NER as it represents name of person in a diverse spectrums, ranging from living things (animals, plants, person) to non-living things (vehicle, furniture). Elson *et al.* employed quoted speech attribution (dialogue and internal monologue) and syntactical approach (adaptation of natural language tools) to identify character in literary fiction, specifically 19th century novels and serials [11]. However, its characters are represented in human alike name.

In this paper, we propose a fully automated named entity recognition framework to overcome the above mentioned issues. Fiction-based domain is used to test on the proposed framework. We study the predefined category of "name of person" but aim to recognize protagonist(s) in fairy tales. Stanford parser and Stanford dependency relation are used to shallowly parse the input file to extract potential NE from the natural text. Word(s) that is/are labeled as VERB between two potential NEs will be extracted to form syntactic triplet structure of subject – verb – object (S-V-O) at a sentence level. WORDNET is then used to substantiate the extracted verb that associates with human action in identifying protagonist. Finally, threshold value is used to filter potential NEs in locating protagonist(s). Part of the work of this paper is a replication and extension of previous research on ontology construction in fiction-based domain [12].

The outline of this paper is as follow: Section 2 discusses the technologies background for this work. Section 3 presents the proposed system framework. Section 4 describes the experiments and the paper is concluded in Section 5.

2 Technologies Background

2.1 Stanford Parser

Stanford parser is a probabilistic parser that analyses syntactic structure of natural language sentences. It has the performance of 86.36% of accuracy in parsing [13]. It is implemented in Java by Stanford University's Natural Language Processing Group and it is available in four languages (English, Chinese, Arabic and German). In this project, English is used to run and test on the selected plain text input.

The parser can read various forms of plain text input and return various analysis formats, including part-of-speech tagged text, phrase structure trees, and a grammatical relations (typed dependency) format. In this work, only phrase structure trees and grammatical relations are used.

2.1.1 Phrase Structure Parse

Phrase structure trees utilized unlexicalized probabilistic context free grammar (PCFG) to achieve greater efficiency and accuracy in parsing sentences. Generally, phrase structure tree is a syntactical structure of sentence that segment group of words into phrases to form the subject and object of the verb.

2.1.2 Stanford Dependencies

Stanford dependencies of English sentences are brought forth based on rules / patterns and Treebank representation from the generated phrase structure trees. There are forty-eight grammatical relations altogether to form the Stanford dependencies. It is a predicate argument representation with a grammatical relation used to bind the right dependencies of two tokens.

In our work, Stanford dependencies is used to identify pair of words that are adjacent to each other based on the index tagged next to it or words that tagged with grammatical relation of "*nn*" will be extracted to form a list of term. This is extremely useful to limit the generation of candidate terms from phrase structure trees.

Our hypotheses for detecting candidate terms are described as follows:

(i) Pair of words that are adjacent often partially contributes to entity recognition.
(ii) Pair of words that are tagged with "*nn*" grammatical relation often denotes important keywords for a domain.

2.2 WordNet

Wordnet[1] is the product of a research project at Princeton University which has attempted to model the lexical knowledge of a native speaker of English [14]. A derivationally related form (DRF) is one of the features available in WordNet being used to identify verb that associates with human action. In this work, each extracted verb (V) that formed S-V-O serves as a keyword for retrieving its corresponding senses' description in derivationally related forms. Each returned description will be examined sentence by sentence. In the presence of either one of the three key phrases of "*someone*", "*a person*" or "*one who*" in the sentence, the verb is considered to be associated with human action.

3 System Framework

The system framework for our proposed NER, focused solely in identifying fiction protagonist(s) is depicted in Fig. 1. The prototypical implementation of the automated NER (protagonist(s)) illustrated in Fig. 1 is explained as below:

The terms used in the framework are:

$term_{sd}$: Terms that are extracted from stanford dependencies based on two criteria; (1) words that are adjacent to each other and (2) words which are tagged with the "nn" grammatical relation
$term_{psp}$: Terms that are tagged with noun phrase (NP) in phrase dependency parse
$NE_{candicate}$: Candidate NE
$VERB_{per}$: Verb that associates with the human action

The first four steps of system framework in this work are generally similar to previous work done in ontology construction for fiction-based domain [12].

Input : Fiction web page
Step 1 : Document cleaning

[1] http://wordnet.princeton.edu/

Eight fairy tales are used to test on the proposed framework. Each fairy tale web page retrieved from the selected domain web pages is cleaned automatically using HTML Context Extractor[2] in order to get rid of non-text content (banner, audio, video, images). A pure text file (.txt) is produced at the end of the cleaning process.

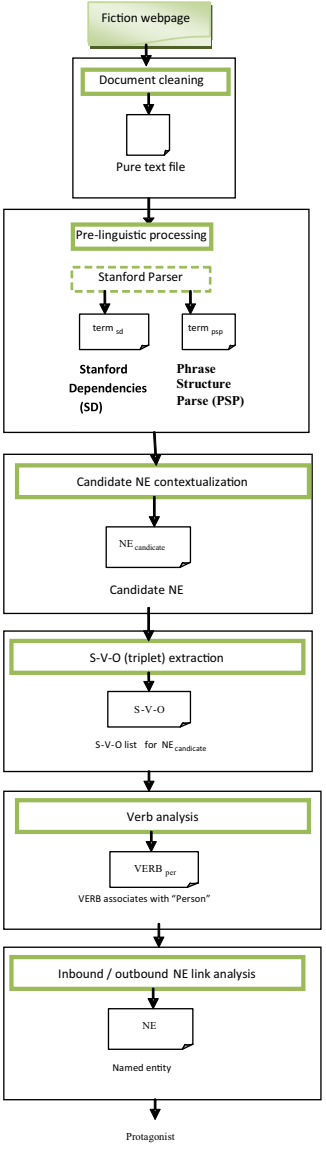


Fig. 1. System framework for NE (protagonists/main actor) recognition

[2] http://senews.sourceforge.net/KCE_README.html

Step 2 : Pre-linguistic processing

Two features available in Stanford parser which are phrase structure parse (PSP) and stanford dependencies (SD) will be used to shallowly examine the text content of the generated pure text file. Parse tree generated by PSP will be further manipulated by extracting phrases that tagged with "*NP*" to form list of term known as $term_{psp}$ whereas predicate argument structure produced by SD will be analyzed according to the two hypotheses mentioned in section 2.1.2 to form list of term known as $term_{sd}$.

Step 3 : Candidate NE Contextualization

NE often appears to be the subject and/or object of a sentence and it usually tagged as "*NP*" in a parse tree. However, not all "*NP*" correspond to NE. For instance, in this work, "*this*", and "*nothing*" are the $term_{psp}$ extracted from "*The Story of Snow White*". Therefore, instead of solely extracting all "*NP*" listed in PSP to form candidate NE. Nested or exact wording of $term_{sd}$ against $term_{psp}$ at sentence level is used to overcome the over generation of candidate NE as illustrated in equation (1) and describe elsewhere[12]. It implies parallelism in generating candidate NE.

$$term_{sd} \cap term_{psp} = NE_{candicate} \quad (1)$$

The overlap between $term_{sd}$ and $term_{psp}$ will result in utilizing $term_{psp}$ to form $NE_{candidate}$. This is due to NE may consist of more than two words while $term_{sd}$ always represents its grammatical relation in two words.

Step 4 : S-V-O (triplet) Extraction

A syntactic triplet structure of S-V-O denotes an event / action being take place between a subject (S) and an object (O). In this work, S implies 1st $NE_{candicate}$ and O marks 2nd $NE_{candicate}$. V is a Verb Phrase (VP) that exist between 1st $NE_{candicate}$ and 2nd $NE_{candicate}$ in a sentence basis. A sentence might have more than one S-V-O syntactic triplet structure. Extracted S can also be the O for another extracted triplet and vice versa.

Step 5 : Verb Analysis

"getDerivationallyRelatedForms" is one of the methods freely available in WordNet API (JAWS)[3] which is used to automatically examine against each verb that resides in the extracted triplets in the previous step. Each verb serves as a keyword for retrieving its corresponding senses' description in derivationally related forms. Key phrases of "*someone who*", "*one who*" or "*a person*" are the hints use to identify verb that associates with human activity. Therefore, each return description will be examined sentence by sentence to locate the

[3] http://lyle.smu.edu/~tspell/jaws/index.html

above mentioned hints. An integer value of 1 will be assigned to each S and O if the verb connected between them contains any of the three hints mentioned. At the end, each S and O might has a value of zero or more for its inbound link and outbound link that associated with human action related verb ($VERB_{per}$), as shown in Fig. 2.

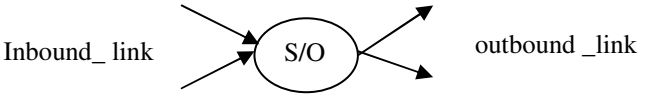


Fig. 2. Inbound and outbound links of S and O

Step 6 : Inbound/Outbound NE Link Analysis

Each inbound link and outbound link of $NE_{candicate}$ will be calculated for its proportional value as shown in equation (2) and (3). A filtering process of $NE_{candicate}$ is then done based on the calculated proportion. $NE_{candicate}$ which has value of 0, 1 and "#DIV/0!" for *in* and/or *out* will be discarded for further analysis in identifying protagonist. 0 means none of the inbound and/or outbound link is associated with human action, 1 denotes all inbound and/or outbound links are associated with human action and "#DIV/0!" shows division by zero, which denotes that no inbound or outbound link attached to $NE_{candicate}$. Later, normalization process of each filtered $NE_{candicate}$ will be performed based on three equations (4), (5) and (6). Equations (4), (5) and (6) imply the inbound link (Nin), outbound link (Nout) and frequency (Nfreq) respectively. Frequency signifies the sum of inbound and outbound links for each filtered $NE_{candicate}$. Finally, each calculated proportion indicates the weight carried by filtered $NE_{candicate}$. High proportion of these three measures may increase the likelihood that the filtered $NE_{candicate}$ is the protagonist of the investigated fairy tale.

$$\text{in} = \text{inbound\_link\_VERB}_{per} / \text{inbound\_link} \tag{2}$$

$$\text{out} = \text{outbound\_link\_VERB}_{per} / \text{outbound\_link} \tag{3}$$

$$\text{Nin}_i = \frac{\text{in}_i}{\sum_{i=1}^{N} \text{in}_i} \tag{4}$$

$$\text{Nout}_i = \frac{\text{out}_i}{\sum_{i=1}^{N} \text{out}_i} \tag{5}$$

$$\text{Nfreq}_i = \frac{\text{inbound\_link}_i + \text{outbound\_link}_i}{\sum_{i=1}^{N} (\text{inbound\_link}_i + \text{outbound\_link}_i)} \tag{6}$$

where 1 <= i <= N and N is the total number of filtered $NE_{candicate}$ in a fairy tale; Nin and Nout are the normalized values for inbound and outbound links respectively while Nfreq is the normalized value for frequency.

Output : Each proportion measured in the previous step is summed up to produce a unit measurement for each filtered $NE_{candicate}$ (equation (7)). Later, median of all weights is calculated to serve as a threshold value in identifying protagonist(s). Therefore, different fairy tales may have different threshold values. Finally, list of protagonist(s) is produced.

$$\text{weight} = \text{Nin} + \text{Nout} + \text{Nfreq} \quad (7)$$

4 Experiments and Discussions

4.1 Dataset

Eight fairy tales from http://www.kidsgen.com were used to test on our proposed framework and the word count for each fairy tales is presented in the second column of Table 1 [1]. The eight fairy tales were chosen as it reflects the aim of this work which is to identify protagonist(s) in diverse spectrums. Some of the protagonists have a character name while some are just the type of the animal/inserts.

4.2 Results and Discussion

The most challenging issue in NER, particularly in the fiction domain of fairy tales lies in its evaluation with gold standard as protagonist name might (1) slightly vary according to the version of the tales or (2) context sensitive to the local flavor. Therefore, a simple survey was conducted on the eight studied fairy tales on 6 primary school students. The third column in Table 1 shows protagonists for each fairy tales obtained from the survey. This result is used as a gold standard in our work to measure the performance of our approach and other NER tools. Three evaluation metrics, namely recall, precision and F-measure are used to evaluate the outcome of the extracted protagonist(s).

Table 1. Fairy tales word count and protagonists

Fairy Tale	Word Count	Protagonist
The Story of Snow White	1913	Snow White
Cinderella	1077	Cinderella
Beauty and the Beast	1357	Beauty, Beast
Rapunzel	1393	Rapunzel
Thumbelina	4348	Tiny
Ugly Duckling	841	Duckling
Sleeping Beauty	1317	Briar Rose
Ant and the Grasshopper	142	Ant, Grasshopper

Table 2 shows the protagonists extracted by our method using 2, 3 and 4 variables. The actual protagonist for each fairy tale appears in bold. 2-variable considered only proportion of inbound link (equation 4) and outbound link (equation 5) that attached to filtered $NE_{candicate}$. 3-variable is the method we have used in work as shown in Step 6 above while 4-variable is an extension of 3-variable with an additional proportion of

links (inbound and outbound link) that associates with $VERB_{per}$ against all links (inbound and outbound link) of each filtered $NE_{candicate}$. Normalization is performed on each calculated proportion. The number of extracted protagonist(s) is the same across 2, 3 and 4 variables. However, the protagonist that were extracted are slightly different between 2-variable and 3, 4-variable. 3-variable and 4-variable extract the same list of protagonists.

Table 2. Inbound/Outbound NE link analysis using different variables

Fairy Tale	2-variable	3-variable	4-variable
The Story of Snow White	**Snow White**	**Snow White**	**Snow White**
Cinderella	**Cinderella**	**Cinderella**	**Cinderella**
Beauty and the Beast	Merchant, Daughter, Horse	Merchant, Daughter, **Beauty**	Merchant, Daughter, **Beauty**
Rapunzel	Time, Enchantress	**Rapunzel**, Enchantress	**Rapunzel**, Enchantress
Thumbelina	Leaf, Mole, **Tiny**, Feather, Earth, Country, Heart	Leaf, Mole, **Tiny**, Feather, Earth, Flower, Bird	Leaf, Mole, **Tiny**, Feather, Earth, Flower, Bird
Ugly Duckling	null	null	Null
Sleeping Beauty	**Briar Rose**	**Briar Rose**	**Briar Rose**
Ant and the Grasshopper	**Ant**	**Ant**	**Ant**

As mentioned in step 6, it is insensible to naively accept protagonist which the $NE_{candicate}$ has only one inbound and/or outbound link, and the verb is associated with human action. Comparatively, $NE_{candicate}$ that has more than one inbound and/or outbound links will definitely have decreased verb probability associates with human action. In fact, protagonists are the main character(s) that should actively engage in story flow. From our experimental dataset, it is observed that there are at least two existences of inbound and/or outbound link for each corresponding $NE_{candicate}$. Finally, filtered $NE_{candicate}$ is produced after eliminating $NE_{candicate}$ that has the value of 0, 1 and "#DIV/0!" for its corresponding *in* and *out*. The above mentioned scenario reflects the approach applied in 2-variable, high proportional value of inbound and outbound link as approach taken in might not sufficient in identifying protagonist in fairy tales. The additional proportional value in 4-variable does not contribute to protagonist identification as it shows the same result as 3-variable. The engagement of filtered $NE_{candicate}$ in a story is shown explicitly by frequency (total link of inbound and outbound). High proportional value of each variable (3-variable) increases the probability of filtered $NE_{candicate}$ to be chosen as protagonist whereas high frequency with low proportional value of inbound and outbound link may prevent $NE_{candicate}$ to be protagonist.

Table 3 compares the results between our approach with three freely available tools on the internet, namely AlchemyAPI[4], General Architecture for Text Engineering

[4] http://www.alchemyapi.com/api/entity/

(GATE)[5] and Illinois named entity tagger[6]. Note that the result on "*Ugly Duckling*" is not included as none of the tools, including our approach is able to identify any protagonist in the story. AlchemyAPI employs hybrid approach in NER where statistical algorithms are combined with natural language processing technology to analyze and identify hundreds of entity types and "people" is one of its types. Contextual cues are used to disambiguate among entity types. For instance, information on a person's career and where they are located are some of the contextual cues used to disambiguate "people" entity type. GATE uses predefined gazetteer list (ANNIE) and rule based approach (JAPE) for finding entity types. Various machine learning techniques can also be imposed on GATE to increase the performance of the NER. However, in this paper, the default GATE NER was used. Illinois extracts NE using external knowledge (gazetteers) and machine learning paradigm. Portability, scalability and no training corpus are the main reasons of choosing these three tools for comparison.

Table 3. Comparison of performance metrics with other tools

		Recall	Precision	F-measure
Beauty and the beast	AlchemyAPI	0.0000	0.0000	0.0000
	GATE	0.0000	0.0000	0.0000
	Illinois	0.5000	1.0000	0.6667
	Our approach	0.5000	0.3333	0.4000
Cinderella	AlchemyAPI	1.0000	0.3333	0.5000
	GATE	0.0000	0.0000	0.0000
	Illinois	1.0000	0.5000	0.6667
	Our approach	1.0000	1.0000	1.0000
Rapunzel	AlchemyAPI	1.0000	1.0000	1.0000
	GATE	0.0000	0.0000	0.0000
	Illinois	0.0000	0.0000	0.0000
	Our approach	1.0000	1.0000	1.0000
Sleeping beauty	AlchemyAPI	1.0000	1.0000	1.0000
	GATE	0.0000	0.0000	0.0000
	Illinois	1.0000	1.0000	1.0000
	Our approach	1.0000	1.0000	1.0000
Snow white	AlchemyAPI	1.0000	1.0000	1.0000
	GATE	1.0000	0.3333	0.5000
	Illinois	0.0000	0.0000	0.0000
	Our approach	1.0000	1.0000	1.0000
Thumbelina	AlchemyAPI	0.0000	0.0000	0.0000
	GATE	0.0000	0.0000	0.0000
	Illinois	1.0000	1.0000	1.0000
	Our approach	1.0000	0.1429	0.2500
Ant and grasshopper	AlchemyAPI	0.0000	0.0000	0.0000
	GATE	0.0000	0.0000	0.0000
	Illinois	0.0000	0.0000	0.0000
	Our approach	0.5000	1.0000	0.6667

[5] http://gate.ac.uk/

[6] http://cogcomp.cs.illinois.edu/page/demo_view/8

Table 4. Average performance results for each tool

	Recall	Precision	F-measure
AlchemyAPI	50.00	42.67	43.75
GATE	12.50	4.17	6.25
Illinois	43.75	43.75	41.67
Our approach	75.00	68.45	66.46

In this work, protagonist can be generally divided into two categories, namely protagonist name (e.g., Snow White and Beauty) and protagonist entity, e.g., insects (ant, grasshopper), and animal (duckling). As can be seen in Table 3, most of the tools perform well on the protagonist name, except for GATE which is constrained by the limited number of name listed in gazetteer. The good performance in this aspect is due to protagonist name is very similar to human name. However, the three tools performed poorly and in fact none of the protagonist entity were identified. Comparatively, our approach is able to perform across the two mentioned categories. In addition to producing comparable performance results with the chosen three tools on the protagonist name, our approach outperforms the rest in identifying protagonist entity, which is insects, *ant* specifically in the fairy tale of "*Ant and grasshopper*". However, another protagonist entity of *grasshopper* is not identifiable as it has 0 value for *in* and that reduced the weight shown in equation (7) to be below the threshold. The same applies to the fairy tale of "*Ugly duckling*" that none of the tools is capable in identifying "*duckling*" as its protagonist because the word "*duckling*" does not seem to be human related name or appearing in the listed gazetteer. Our approach failed too, owing to the 0 value generated for *in* (equation 4). There is only one action imposed on "*duckling*" and the action is not related to human action. A protagonist should interact actively in story flow and contribute to inbound link ($VERB_{per}$, action being taken towards protagonist) and outbound link ($VERB_{per}$, action taken by protagonist). Therefore, both number of actions being taken and imposed on filtered $NE_{candicate}$, and its relevancy to human action give strong impact during protagonist identification. Lacking of either factor may hamper the effort of protagonist identification.

File size and activities/events affiliated with protagonist in fairy tale do impact the performance results of protagonist identification. This is due to the $Verb_{per}$ for inbound and outbound link will influence the proportion of $NE_{candicate}$'s inbound and outbound link. Small file size imposes limited activities or events affiliated with protagonist, which reduces the probability of inbound and outbound link that contain $VERB_{per}$. This can be seems very clearly for the fairy tales of "*Ant and grasshopper*" and "*Ugly duckling*" which have 142 and 841 word count respectively.

Recall, precision and F-measure are interdependent. High recall with low precision and vice versa might yield low F-measure, while high recall and high precision will definitely generate high F-measure. With the existence of one or two protagonists for a fairy tale always incur low precision if the number of identifiable filtered $NE_{candicate}$ is high and vice versa. Therefore, carefully taking care of each fairy tales nature is likely to improve the performance results of the protagonist identification. Table 4 summarizes the comparative study. Our approach yields better results compared to the other three tools.

5 Conclusion

This paper presents an algorithmic framework for protagonist identification in fiction domain. Comparatively, our proposed method is able to perform consistently. For future work, we intend to improve the protagonist identification by collaborating with VerbNet [15], which is the largest on-line verb lexicon in English that incorporates both semantic and syntactic about its content.

References

1. Chiticariu, L., Krishnamurthy, R., Li, Y.Y., Reiss, F., Vaithyanathan, S.: Domain Adaptation of Rule-Based Annotators for Named-Entity Recognition Tasks. In: Empirical Methods in Natural Language Processing, Massachusetts, pp. 1002 – 1012 (2010)
2. McCallum, A., Li, W.: Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons. In: 7th Conference on Natural Language Learning, pp. 188–191 (2003)
3. Klein, D., Smarr, J., Nguyen, H., Manning, C.D.: Named Entity Recognition with Character-Level Models. In: 7th Conference on Natural Language Learning, pp. 180–183 (2003)
4. Irmak, U., Kraft, R.: A Scalable Machine-Learning Approach for Semi-Structured Named Entity Recognition. In: 19th International World Wide Web Conference, North Carolina, pp. 461–470 (2010)
5. Le, H.T., Nguyen, T.H.: Name Entity Recognition using Inductive Logic Programming. In: Symposium on Information and Communication Technology, Vietnam, pp. 71–77 (2010)
6. Minkov, E., Wang, R.C., Cohen, W.W.: Extracting Personal Names from Email: Applying Named Entity Recognition to Informal Text. In: Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Vancouver, pp. 443–450 (2005)
7. Sekine, S., Nobata, C.: Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy. In: 4th International Conference on Language Resource and Evaluation (LREC), pp. 1977–1980 (2004)
8. Smarr, J., Manning, C.D.: Classifying Unknown Proper Noun Phrases without Context. Technical Report dbpubs/2002-46. Stanford University, Stanford, CA (2002)
9. Artiles, J., Amigo, E., Gonzalo, J.: The Role of Named Entities in Web People Search. In: Conference on Empirical Methods in Natural Language Processing, Singapore, pp. 534–542 (2009)
10. Fleischman, M., Hovy, E.: Fine Grained Classification of Named Entities. In: 19th International Conference on Computational Linguistics, pp. 1–7 (2002)
11. Elson, D.K., Dames, N., McKeown, K.R.: Extracting Social Networks from Literary Fiction. In: 48th Annual Meeting of the Association for Computational Linguistic, Uppsala, Sweden, pp. 138–147 (2010)
12. Goh, H.N., Kiu, C.C., Soon, L.K., Ranaivo, B.: Automatic Ontology Construction in Fiction-based Domain. International Journal of Software Engineering and Knowledge Engineering (2011) (in Press)
13. Klein, D., Manning, C.D.: Accurate Unlexicalized Parsing. In: 41st Meeting of the Association for Computational Lingusitic, pp. 423–430 (2003)
14. Stark, M.M., Riesenfeld, R.F.: WordNet: An Electronic Lexical Database. In: 11th Eurographics Workshop on Rendering (1998)
15. Verbnet, `http://verbs.colorado.edu/~mpalmer/projects/verbnet.html`

CD: A Coupled Discretization Algorithm

Can Wang[1], Mingchun Wang[2], Zhong She[1], and Longbing Cao[1]

[1] Centre for Quantum Computation and Intelligent Systems
Advanced Analytics Institute, University of Technology, Sydney, Australia
{canwang613,zhong2024,longbing.cao}@gmail.com
[2] School of Science, Tianjin University of Technology and Education, China
mchwang123@163.com

Abstract. Discretization technique plays an important role in data mining and machine learning. While numeric data is predominant in the real world, many algorithms in supervised learning are restricted to discrete variables. Thus, a variety of research has been conducted on discretization, which is a process of converting the continuous attribute values into limited intervals. Recent work derived from entropy-based discretization methods, which has produced impressive results, introduces information attribute dependency to reduce the uncertainty level of a decision table; but no attention is given to the increment of certainty degree from the aspect of positive domain ratio. This paper proposes a discretization algorithm based on both positive domain and its coupling with information entropy, which not only considers information attribute dependency but also concerns deterministic feature relationship. Substantial experiments on extensive UCI data sets provide evidence that our proposed coupled discretization algorithm generally outperforms other seven existing methods and the positive domain based algorithm proposed in this paper, in terms of simplicity, stability, consistency, and accuracy.

1 Introduction

Discretization is probably one of the most broadly used pre-processing techniques in machine learning and data mining [6,13] with various applications, such as solar images [2] and mobile market [14]. By using discretization algorithms on continuous variables, it replaces the real distribution of the data with a mixture of uniform distributions. Generally, discretization is a process that transforms the values of continuous attributes into a finite number of intervals, where each interval is associated with a discrete value. Alternatively, this process can be also viewed as a method to reduce data size from huge spectrum of numeric variables to a much smaller subset of discrete values.

The necessity of applying discretization on the input data can be due to different reasons. The most critical one is that many machine learning and data mining algorithms are known to produce better models by discretizing continuous attributes, or only applicable to discrete data. For instance, rule extraction techniques with numeric attributes often lead to build rather poor sets of rules [1]; it is not always realistic to presume normal distribution for the continuous values to enable the Naive Bayes classifier to estimate the frequency probabilities

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 407–418, 2012.

[13]; decision tree algorithms cannot handle numeric features in tolerable time directly, and only carry out a selection of nominal attributes [9]; and attribute reduction algorithms in rough set theory can only apply to the categorical values [10]. However, real-world data sets predominantly consist of continuous or quantitative attributes. One solution to this problem is to partition numeric domains into a number of intervals with corresponding breakpoints. As we know, the number of different ways to discretize a continuous feature is huge [6], including binning-based, chi-based, fuzzy-based [2], and entropy-based methods [13], etc. But in general, the goal of discretization is to find a set of breakpoints to partition the continuous range into a small number of intervals with high distribution stability and consistency, and then to obtain a high classification accuracy. Thus, different discretization algorithms are evaluated in terms of four measures: *simplicity* [5], *stability* [3], *consistency* [6], and *accuracy* [5,6].

Of all the discretization methods, the entropy-based algorithms are the most popular due to both their high efficiency and effectiveness [1,6], including *ID3*, *D2*, and *MDLP*, etc. However, this group of algorithms only concern the decrease of uncertainty level by means of information attribute dependency in a decision table [5], which is not rather convincing. From an alternative perspective, we propose to improve the discretization quality by increasing the certainty degree of a decision table in terms of deterministic attribute relationship, which is revealed by the positive domain ratio in rough set theory [10]. Furthermore, based on the rationales presented in [8,12], we take into account both the decrement of uncertainty level and increment of certainty degree to induce a Coupled Discretization (CD) algorithm. This algorithm selects the best breakpoint according to the importance function composed of the information entropy and positive domain ratio in each run. The key contributions are as follows:

- Consider the information and deterministic feature dependencies to induce the coupled discretization algorithm in a comprehensive and reasonable way.
- Evaluate our proposed algorithm with existing classical discretization methods on a variety of benchmark data sets from internal and external criteria.
- Develop a way to define the importance of breakpoints flexibly with our fundamental building blocks according to specific requirements.
- Summarize a measurement system, including *simplicity*, *stability*, *consistency*, and *accuracy*, to evaluate discretization algorithm completely.

The paper is organized as follows. Section 2 briefly reviews the related work. In Section 3, we describe the problem of discretization within a decision table. Discretization algorithm based on information entropy is specified in Section 4. In Section 5, we propose the discretization algorithm based on positive domain. Coupled discretization algorithm is presented in Section 6. We conduct extensive experiments in Section 7. Finally, we end this paper in Section 8.

2 Related Work

In earlier days, simple methods such as Equal Width (EW) and Equal Frequency (EF) [6] are used to discretize continuous values. Afterwards, the technology for

discretization develops rapidly due to the great need for effective and efficient machine learning and data mining methods. From different perspectives, discretization methods can be classified into distinct categories. A global method uses the entire instance space to discretize, including *Chi2* and *ChiM* [6], etc.; while a local one partitions the localized region of the instance space [5], for instance, *1R*. Supervised discretization considers label information such as *1R* and *MDLP* [1]; however, unsupervised method does not, e.g., *EW*, *EF*. Splitting method such as *MDLP* proceeds by keeping on adding breakpoints, whereas the merging approach by removing breakpoints obtains bigger intervals, e.g., *Chi2* and *ChiM*. The discretization method can also be viewed as dynamic or static by considering whether a classifier is incorporated during discretization, for example, *C4.5* [6] is a dynamic way to discretize continuous values when building the classifier. The last dichotomy is direct vs. incremental, while direct method needs the pre-defined number of intervals, including *EW* and *EF*; incremental approach requires an additional criterion to stop the discretization process, such as *MDLP* and *ChiM* [3]. In fact, our proposed method *CD* is a global-supervised-splitting-incremental algorithm, and comparisons with the aforementioned classical methods are conducted in Section 7.

3 Problem Statement

In this section, we formalize the discretization problem within a decision table, in which a large number of data objects with the same feature set can be organized.

A *Decision Table* is an information and knowledge system which consists of four tuples $(U, C \bigcup D, V, f)$. $U = \{u_1, \cdots, u_m\}$ is a collection of m objects. $C = \{c_1, \cdots, c_n\}$ and D are condition attribute set and decision attribute set, respectively. V_C is a set of condition feature values, V_D is a set of decision attribute values, and the whole value set is $V = V_C \bigcup V_D$. $f : U \times (C \bigcup D) \to V$ is an information function which assigns every attribute value to each object. $D \neq \varnothing$ if there is at least one decision feature $d \in D$. The entry x_{ij} is the value of continuous feature c_j $(1 \leq j \leq n)$ for object u_i $(1 \leq i \leq m)$. If all the condition attributes are continuous, then we call it a *Continuous Decision Table*.

Let $S = (U, C \bigcup D, V, f)$ be a continuous decision table, $S(P) = (U, C^* \bigcup D, V^*, f^*)$ is the *Discretized Decision Table* when adding breakpoint set P, where C^* is the discretized condition attribute, V^* is the attribute value set composed of discretized values V_C^* and decision value V_D, and $f^* : U \times (C^* \bigcup D) \to V^*$ is the discretized information function. For simplicity, we consider only one decision attribute $d \in D$. Below, a consistent discrete decision table is defined:

Definition 1. *A discrete decision table $S(P) = (U, C^* \bigcup D, V^*, f^*)$ is **consistent** if and only if any two objects have identical decision attribute value when they have the same condition attribute values.*

In fact, the discretization of a continuous decision table S is the search of a proper breakpoint set P, which makes discretized decision table $S(P)$ consistent. In this process, different algorithms result in distinct breakpoint sets, thus correspond to

various discretization results. Chmielewski and Grzymala-Busse [5] suggest three guidelines to ensure successful discretization, that is complete process, simplest result and high consistency. Thus, among all the breakpoints, we strive to obtain the smallest set of breakpoints which make the least loss on information during discretization.

4 Discretization Algorithm Based on Information Entropy

In this section, we present a discretization method which uses class information entropy to evaluate candidate breakpoints in order to select boundaries [6]. The discretization algorithm based on entropy (IE) is associated with the information gain of objects divided by breakpoints to measure the importance of them.

Definition 2. *Let $W \subseteq U$ be the subset of objects which contains $|W|$ objects. k_t denotes the number of the objects whose decision attribute values are $y_t (1 \leq t \leq |d|)$, where $|d|$ is the number of distinct decision values. Then the* ***class information entropy*** *of W is defined as follows:*

$$H(W) = -\sum_{t=1}^{|d|} p_t \log_2 p_t, \ \textit{where}\ p_t = \frac{k_t}{|W|} \tag{4.1}$$

Note that $H(W) \geq 0$. Smaller $H(X)$ corresponds to lower uncertainty level of the decision table [5,6], since some certain decision attribute values play the leading role in object subset W. In particular, $H(W) = 0$ if and only if all the objects in subset W have the same decision attribute value.

For a discretized decision table $S(P)$, let $W_1, W_2, \cdots, W_r$ be the sets of equivalence classes based on the identical condition attribute values. Then, the class information entropy of the discretized decision table $S(P)$ is defined as $H(S(P)) = \sum_{i=1}^{r} \frac{|W_i|}{|U|} H(W_i)$. Based on Definition 1, we obtain the relationship between entropy and consistency as follows. The proof is shown in the Appendix.

Theorem 1. *A discretized decision table $S(P)$ is consistent if $H(S(P)) = 0$.*

After the initial partition, $H(S(P))$ is usually not equal to 0, which means $S(P)$ is not consistent. Accordingly, we need to select breakpoints from candidate set $Q = \{q_1, q_2, \cdots, q_l\}$, and it is necessary to measure the importance of every element of Q to determine which one to choose in the next step. Let $S(P \bigcup \{q_i\})$ be the discretized decision table when inserting the breakpoint set $P \bigcup \{q_i\}$ to the continuous decision table S, and the corresponding class information entropy is $H(S(P \bigcup \{q_i\}))$. The existing standard [6] to measure the importance of breakpoint q_i is defined as:

$$H(q_i) = H(S(P)) - H(S(P \bigcup \{q_i\})). \tag{4.2}$$

Note that the greater the decrease $H(q_i)$ of entropy, the more important the breakpoint q_i. Since $H(S(P))$ is a constant value for every $q_i (1 \leq i \leq l)$, then the smaller the entropy $H(S(P \bigcup \{q_i\}))$, the larger probable the breakpoint q_i will be chosen.

5 Discretization Algorithm Based on Positive Domain

Alternatively, we propose another discretization method incorporated with rough set theory to select breakpoints to partition the continuous values. The discretization algorithm based on positive domain (PD) is built upon the indiscernibility relations induced by the equivalence classes to evaluate the significance of the breakpoints. Firstly, we recall the relevant concept in rough set theory [10].

Definition 3. *Let U be a universe, P, Q are the equivalence relations over set U, then the Q* ***positive domain*** *(or* ***positive region****) of P is defined as:*

$$POS_P(Q) = \bigcup_{W \in U/Q} \{u : u \in U \wedge [u]_P \subseteq W\}, \tag{5.1}$$

where $W \in U/Q$ is the equivalence class based on relation Q, $[u]_P$ is the equivalence class of u based on relation P.

In the discretized decision table $S(P) = (U, C^* \bigcup D, V^*, f^*)$, let C^* be the equivalence relation of "two objects have the same condition attribute values", let D denote the equivalence relation of "two object have the same decision attribute value". Then, the positive domain ratio of the decision table $S(P)$ is $R(S(P)) = |POS_{C^*}(D)|/|U|$. Note that $|U|$ is the number of objects, and $0 \leq R(S(P)) \leq 1$. The greater the ratio $R(S(P))$, the higher the certainty level of discretized decision table [8,10]. Below, we reveal the consistency condition for the PD algorithm. The proof is also shown in the Appendix.

Theorem 2. *A discretized decision table $S(P)$ is consistent if $R(S(P)) = 1$.*

Similarly, we usually have $R(S(P)) \neq 1$, that is to say, $S(P)$ is not consistent after initialization. Thus, it is necessary to choose breakpoints from candidate set $Q = \{q_1, q_2, \cdots, q_l\}$ according to the significance order of all the candidate breakpoints for the next insertion. Let $R(S(P \bigcup \{q_i\}))$ denote the positive domain ratio of the discretized decision table $S(P \bigcup \{q_i\})$. We could then define the importance of breakpoint q_i as:

$$R(q_i) = R(S(P \bigcup \{q_i\})) - R(S(P)). \tag{5.2}$$

Note that the larger the increase $R(q_i)$ of ratio, the greater importance of the breakpoint q_i. Since $R(S(P))$ is a constant for each candidate $q_i (1 \leq i \leq l)$, therefore, the larger the ratio $R(S(P \bigcup \{q_i\}))$, the more important this breakpoint q_i.

6 Discretization Algorithm Based on the Coupling

Discretization algorithms are considered in terms of information entropy and positive domain in Section 4 and Section 5, respectively. In a discretized decision table, the information entropy measures the uncertainty degree from the

perspective of information attribute relationship, while the positive domain ratio reveals the certainty level with respect to the deterministic feature dependency [8]. In this Section, we focus on both the information and deterministic attribute dependencies to derive the coupled discretization (CD) algorithm.

Theoretically, Wang et. al [12] compared algebra viewpoint in rough set and information viewpoint in entropy theory. Later on, Chen and Wang [4] applied the aggregation of them to the hybrid space clustering. Similarly, by taking into account both the increment of certainty level and the decrement of uncertainty degree in a decision table, we consider to combine the PD and IE based methods together to get the CD algorithm. This algorithm measures the importance of breakpoints comprehensively and reasonably by aggregating the positive domain ratio function $R(\cdot)$ and the class information entropy function $H(\cdot)$ together. Alternatively, we propose one option to quantify the coupled importance:

Definition 4. *For a discretized decision table $S(P)$, we have the* ***coupled importance*** *of breakpoint set P be:*

$$RH(P, q_i) = k_1 R(q_i) + k_2 H(q_i), \tag{6.1}$$

where $R(q_i)$ and $H(q_i)$ are the importance functions of breakpoint q_i according to (5.2) and (4.2), respectively; $k_1, k_2 \in [0, 1]$ are the corresponding weights.

For every condition attribute $c_j \in C$ in the continuous decision table $S = (U, C \bigcup D, V, f)$, its values are ordered as $l_{c_j} = x'_{1j} < \cdots < x'_{mj} = r_{c_i}$.Then, we define the candidate breakpoint as: $q_{ij} = \frac{x'_{ij}+x'_{i+1,j}}{2} (1 \leq i \leq m-1, 1 \leq j \leq n)$.

The process of the discretization algorithm based on the coupling of positive domain and information entropy is designed as follows. The algorithm below clearly shows that its computational complexity is $O(m^2n^2)$ based on the loops.

7 Experiment and Evaluation

In this section, several experiments are performed on extensive UCI data sets to show the effectiveness of our proposed coupled discretization algorithm. All the experiments are conducted on a Dell Optiplex 960 equipped with an Intel Core 2 Duo CPU with a clock speed of 2.99 GHz and 3.25 GB of RAM running Microsoft Windows XP. For simplicity, we just assign the weights $k_1 = k_2 = 0.5$ in Definition 4 and Algorithm 1.

To the best of our knowledge, there are mainly four dimensions to evaluate the quality [3,5,6] of discretization algorithms as follows:

- Stability: How to measure the overall spread of the values in each interval.
- Simplicity: The fewer the break points, the better the discretization result.
- Consistency: The inconsistencies caused by discretization should not be large.
- Accuracy: How discretization helps improve the classification accuracy.

Discretization methods that adhere to internal criterion assign the best score to the algorithm that produces break points with high *stability* and low *simplicity*;

Algorithm 1. Coupled Algorithm for Discretization

Data: Decision table S with m objects and n attributes (value x_{ij}), and k_1, k_2.
Result: breakpoint set P.
begin
  breakpoint set $P = \emptyset$, candidate breakpoint set $Q = \emptyset$;
  for $j = 1 : n$ **do**
    $\{x'_{ij}\} \leftarrow sort(\{x_{ij}\})$;
  for $j = 1 : n$ **do**
    for $i = 1 : (m-1)$ **do**
      candidate breakpoint $q_{ij} \leftarrow \frac{x'_{ij}+x'_{i+1,j}}{2}$, $Q = \{q\}$;
  Fix the first breakpoint $p_1 \leftarrow argmin_q H(S(P \bigcup \{q_{ij}\}))$;
  while $H(S(P)) \neq 0 \wedge R(S(P)) \neq 1$ **do**
    for *candidate* $k = 1 : |Q|$ **do**
      calculate $RH(P, q_k)$ according to (6.1);
    $q_{max} \leftarrow argmax_q RH(P, q_k)$;
    $P \leftarrow P \cup \{q_{max}\}$, $Q \leftarrow Q \backslash \{q_{max}\}$;
  Output breakpoint set P;
end

while discretization approaches that adhere to external criterion compare the results of the algorithm against some external benchmark, such as predefined classes or labels indicated by *consistency* and *accuracy*. From these two perspectives, the experiments here are divided into two categories according to different evaluation standards: internal criteria (*stability*, *simplicity*) and external criteria (*consistency*, *accuracy*), as shown in Section 7.1 and Section 7.2, respectively.

7.1 Internal Criteria Comparison

With respect to the internal criterion, i.e., stability and simplicity, the goal in this set of experiments is to show the superiority of our proposed coupled discretization (*CD*) algorithm against some classic methods [6] such as Equal Frequency (*EF*), *1R*, *MDLP*, *Chi2*, and Information Entropy-based (*IE*) algorithms.

Specifically, *simplicity* measure is described as the total number of intervals (*NOI*) for all the discretized attributes. More complicatedly, the *stability* measures are constructed from a series of estimated probability distributions for the individual intervals constructed by incorporating the method of Parzen windows [3]. As one of the induced measure, Attribute Stability Index (ASI_j) is constructed from the weighted sum of the Stability Index (SI_{jk}), which describes the value distribution for each interval I_k of attribute c_j. The measure SI_{jk} follows $0 < SI_{jk} < 1$, if SI_{jk} is near 0 then its values are next to the break points of the interval I_k, while SI_{jk} is close to 1 when its values are near the center of the interval I_k. Furthermore, we have $0 < ASI_j < 1$, and the larger the ASI_j value, the more stable and better the discretization method. Here, we adapt this measure to be the Average Attribute Stability Index (*AASI*), which is the weighted sum of ASI_j for all the attributes $c_j (1 \leq j \leq n)$: $AASI = \sum_{j=1}^{n} ASI_j / n$.

The break points and intervals produced by the aforementioned six discretization methods are then analyzed on 15 UCI data sets in different scales, ranging from 106 to 1484 (number of objects). The results are reported in Table 1. As discussed, larger *AASI*, smaller *NOI* indicate more stable and simpler characterization of the interval partition capability, which further corresponds to a better discretization algorithm. The values in bold are the best relevant indexes for each data. From Table 1, we observe that with the exception of only few items (in italic), the other indexes all show that our proposed *CD* algorithm is better than the other five classical approaches (*EF*, *1R*, *MDLP*, *Chi2*, *IE*) in most cases from the perspectives of *stability* and *simplicity*. It is also worth noting that our proposed *CD* always outperforms the *IE* algorithm presented in Section 4 in terms of *stability*, which verifies the benefit of aggregating the positive domain.

Table 1. Discretization Comparison with Stability and Simplicity

Data set	Average Attribute Stability Index						Number of Intervals					
	EF	*1R*	*MDLP*	*Chi2*	*IE*	*CD*	*EF*	*1R*	*MDLP*	*Chi2*	*IE*	*CD*
Tissue	0.57	0.56	0.27	0.64	0.15	**0.68**	81	96	48	38	26	**24**
Echo	0.44	0.52	***0.67***	0.50	0.32	0.65	70	44	17	21	19	**14**
Iris	0.33	0.28	0.66	0.67	0.39	**0.72**	16	17	12	***11***	14	14
Hepa	0.16	0.21	0.21	0.18	0.19	**0.28**	118	54	***18***	34	19	21
Wine	0.59	0.59	0.65	***0.83***	0.60	0.80	169	130	16	24	**13**	**13**
Glass	0.63	0.50	0.80	0.56	0.75	**0.82**	46	86	50	27	34	**20**
Heart	0.25	0.25	0.40	0.31	0.34	**0.51**	70	61	42	43	28	**26**
Ecoli	0.51	0.29	0.62	0.54	0.51	**0.72**	36	76	***27***	33	30	28
Liver	0.66	0.24	0.78	0.69	0.74	**0.79**	30	70	68	74	***22***	24
Auto	0.58	0.35	0.69	0.65	0.67	**0.73**	47	73	39	67	39	**31**
Housing	0.50	0.64	0.72	0.56	0.61	**0.78**	142	32	29	340	25	**13**
Austra	0.28	0.15	0.39	0.32	0.36	**0.41**	83	102	21	98	26	**17**
Cancer	0.17	0.13	***0.27***	0.22	0.22	0.26	44	31	29	40	**18**	**18**
Pima	0.55	0.32	***0.73***	0.60	0.20	0.70	48	161	***24***	35	33	29
Yeast	0.47	0.17	0.62	0.55	0.30	**0.70**	***45***	47	55	51	51	49

7.2 External Criterion Comparison

In this part of our experiments, we focus on the other two aspects of evaluation measures: *consistency* and *accuracy*. Two independent groups of experiments are conducted with extensive data sets based on machine learning applications.

According to Liu et al. [6], *consistency* is defined by having the least pattern inconsistency count which is calculated as the number of times this pattern appears in the data minus the largest number of corresponding class labels. Thus, the fewer the inconsistency count, the better the discretization quality. Based on the discretization results in Section 7.1, we compute the sum of all the pattern inconsistency counts for all possible patterns of the original continuous feature subset. *Consistency* evaluation is conducted on nine data sets with different number of objects, ranging from 132 (Echo) to 768 (Pima) in an increasing

order. We also consider the other seven discretization methods for comparison, i.e., Equal Frequency (EW), EF, $1R$, $MDLP$, $ChiM$, $Chi2$, and IE.

As shown in Fig. 1, the total inconsistency counts of IE and our proposed CD are always 0 on all the data sets, because the stopping criteria are the consistency conditions presented in Theorem 1 and Theorem 2. However, $MDLP$ seems to perform the worst in terms of the *consistency* index, and the inconsistency counts of the other five algorithms fall in the intervals between those of $MDLP$ and CD for all the data sets. These observations reveal the fact that algorithms IE and CD are the most consistent candidates for discretization. While IE and AD both indicate a surprisingly high *consistency*, in general, CD produces higher *stability* (larger $AASI$) and lower *simplicity* (smaller NOI), as presented in Table 1.

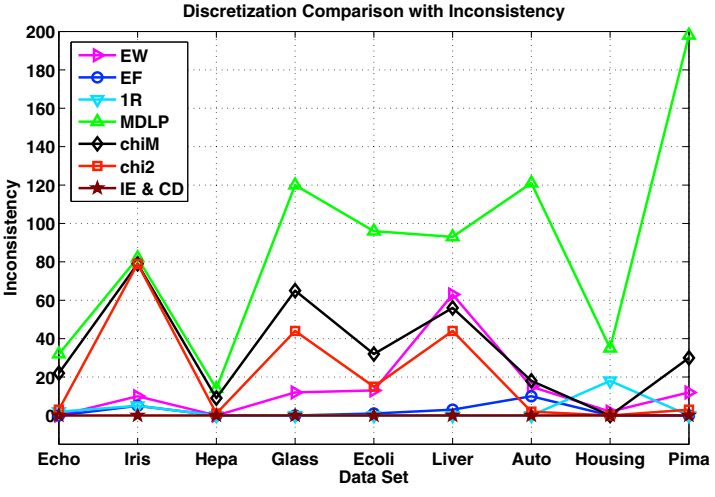


Fig. 1. Discretization Comparison with Consistency

How does discretization affect the classification learning accuracy? As Liu et al. [6] indicate, accuracy is usually obtained by running a classifier in cross validation mode. In this group of experiments, two classification algorithms are taken into account. i.e., Naive-Bayes, and Decision Tree (C4.5). A Naive Bayes (NB) classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions [13]. $C4.5$ is an algorithm used to generate a decision tree (DT) for classification. As pointed out in Section 1, the continuous attributes take too many different values for the NB classifier to estimate frequencies; DT algorithm can only carry out a selection process of nominal features [9]. Thus, discretization is rather critical for the task of classification learning. Here, we evaluate the discretization methods with the classification accuracies induced by NB and $DT(C4.5)$, respectively.

Fig. 2 reports the results on 9 data sets with distinct data sizes, which vary from 150 to 1484 in terms of the number of objects. As can be clearly seen from this figure, the classification algorithms with CD, whether NB or DT, mostly

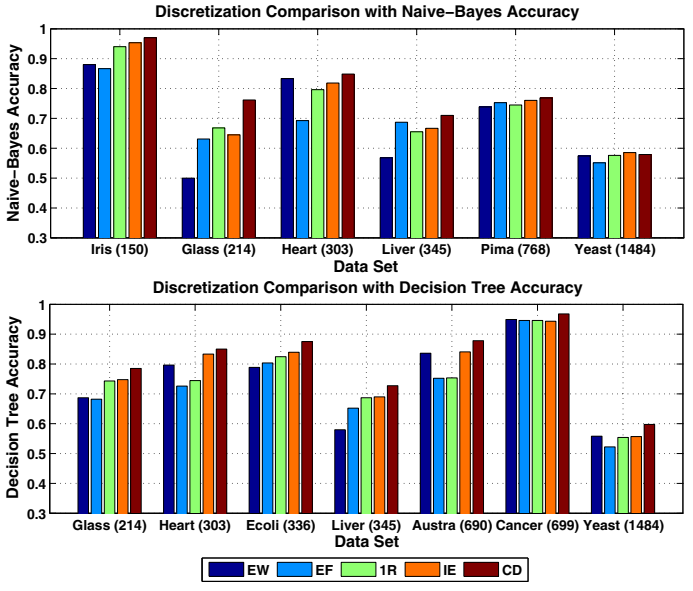


Fig. 2. Discretization Comparison with Accuracy

outperform those with other discretization methods (i.e., *EW*, *EF*, *1R*, *IE*) from the perspective of average *accuracy*. That is to say, discretization algorithm *CD* is better than others on classification qualities. Though for the data set *Yeast*, the average *accuracy* measures induced by *NB* with *CD* are slightly smaller than that with *IE*, the *stability* measures shown in Table 1 indicate that *CD* is better than *IE*. Therefore, our proposed discretization algorithm *CD* is better than other candidates with respect to the classification *accuracy* measure.

Besides, we lead a comparison among the algorithms presented in Section 4 (*IE*), Section 5 (*PD*), and Section 6 (*CD*). Due to space limitations, only *simplicity* and *accuracy* measures are considered to evaluate these three discretization algorithms. Here, we take advantage of the k-nearest neighbor algorithm (*k-NN*) [7], which is a method for classifying objects based on closest training examples in the feature space. After discretization, five data sets are used for classification with both *1-NN* and *3-NN*, in which 70% of the data is randomly chosen for training with the rest 30% for testing. As indicated in Table 2, our proposed *CD* method generally outperforms the existing *IE* algorithm and proposed *PD* algorithm. Specifically for *3-NN*, the average *accuracy* improving rate ranges from 2.35% (Iris) to 27.06% (Glass) when compared *CD* with *IE*. With regard to *1-NN*, this rate falls within −1.58% (Glass) and 1.96% (Austra) between *CD* and *PD*. However, by considering both *simplicity* and *accuracy*, we find out that *CD* is the best one since it takes the aggregation of the other two candidates.

Consequently, we draw the following conclusion: our proposed *Coupled Discretization* algorithm generally outperforms the other classical candidates in terms of all the four measures: *stability*, *simplicity*, *consistency*, and *accuracy*.

Table 2. Comparison between *IE* & *PD* & *CD*

Dataset	Number of Intervals			Accuracy by *1-NN*			Accuracy by *3-NN*		
	IE	*PD*	*CD*	*IE*	*PD*	*CD*	*IE*	*PD*	*CD*
Iris (150)	14	***10***	14	95.24	96.95	**97.48**	94.48	94.10	**95.54**
Glass (214)	34	79	**20**	61.60	***79.53***	78.27	57.73	66.67	**67.12**
Heart (303)	28	45	**26**	63.28	73.33	**74.29**	62.86	75.87	**77.04**
Austra (690)	26	78	**17**	70.14	76.60	**78.10**	73.17	***80.54***	79.96
Pima (768)	33	74	**29**	67.10	70.74	**71.04**	69.33	73.09	**73.12**

8 Conclusion

Discretization algorithm plays an important role in the applications of machine learning and data mining. In this paper, we propose a new global-supervised-splitting-incremental algorithm *CD* based on the coupling of positive domain and information entropy. This method measures the importance of breakpoints in a comprehensive and reasonable way. Experimental results show that our proposed algorithm can effectively improve the distribution *stability* and classification *accuracy*, optimize the *simplicity* and reduce the total *inconsistency* counts. We are currently applying the *CD* algorithm to the estimation of web site quality with flexible weights k_1, k_2 and stopping criteria, and we also consider the aggregation of the *CD* algorithm with coupled nominal similarity [11] to induce coupled numeric similarity and clustering ensemble applications.

Acknowledgment. This work is sponsored by Australian Research Council Grants (DP1096218, DP0988016, LP100200774, LP0989721), and Tianjin Research Project (10JCYBJC07500).

References

1. An, A., Cercone, N.: Discretization of Continuous Attributes for Learning Classification Rules. In: Zhong, N., Zhou, L. (eds.) PAKDD 1999. LNCS (LNAI), vol. 1574, pp. 509–514. Springer, Heidelberg (1999)
2. Banda, J.M., Angryk, R.A.: On the effectiveness of fuzzy clustering as a data discretization technique for large-scale classification of solar images. In: FUZZ-IEEE 2009, pp. 2019–2024 (2009)
3. Beynon, M.J.: Stability of continuous value discretisation: an application within rough set theory. International Journal of Approximate Reasoning 35, 29–53 (2004)
4. Chen, C., Wang, L.: Rough set-based clustering with refinement using Shannon's entropy theory. Computers and Mathematics with Applications 52(10-11), 1563–1576 (2006)
5. Chmielewski, M.R., Grzymala-Busse, J.W.: Global discretization of continuous attributes as preprocessing for machine learning. International Journal of Approximate Reasoning 15, 319–331 (1996)
6. Liu, H., Hussain, F., Tan, C.L., Dash, M.: Discretization: an enabling technique. Data Mining and Knowledge Discovery 6, 393–423 (2002)

7. Liu, W., Chawla, S.: Class Confidence Weighted kNN Algorithms for Imbalanced Data Sets. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part II. LNCS, vol. 6635, pp. 345–356. Springer, Heidelberg (2011)
8. Pawlak, Z., Wong, S.K.M., Ziarko, W.: Rough sets: probabilistic versus deterministic approach. International Journal of Man-Machine Studies 29, 81–95 (1988)
9. Qin, B., Xia, Y., Li, F.: DTU: A Decision Tree for Uncertain Data. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 4–15. Springer, Heidelberg (2009)
10. Son, N.H., Szczuka, M.: Rough sets in KDD. In: PAKDD 2005, pp. 1–91 (2005)
11. Wang, C., Cao, L., Wang, M., Li, J., Wei, W., Ou, Y.: Coupled nominal similarity in unsupervised learning. In: CIKM 2011, pp. 973–978 (2011)
12. Wang, G., Zhao, J., An, J., Wu, Y.: A comparative study of algebra viewpoint and information viewpoint in attribute reduction. Fundamenta Informaticae 68, 289–301 (2005)
13. Yang, Y., Webb, G.I.: Discretization for Naive-Bayes learning: managing discretization bias and variance. Machine Learning 74, 39–74 (2009)
14. Zhang, X., Wu, J., Yang, X., Lu, T.: Estimation of market share by using discretization technology: an application in China mobile. In: ICCS 2008, pp. 466–475 (2008)

Appendix: Theorem Proof

Proof. – [**Theorem 1**] Since $H(S(P)) = 0$, then

$$\frac{|W_1|}{|U|}H(W_1) + \frac{|W_2|}{|U|}H(W_2) + \cdots + \frac{|W_r|}{|U|}H(|W_r|) = 0.$$

Because we have $H(W) \geq 0$, then $H(W_1) = H(W_2) = \cdots = H(W_r) = 0$.

According to the definition of class information entropy of $W_i(i = 1, 2, \cdots, r)$, $H(W_i) = -\sum_{j=1}^{r(d)} p_j \log_2 p_j$. Since $0 \leq p_j \leq 1, \log_2 p_j \leq 0, H(W_i) = 0$, then $p_j = \frac{k_j}{|W_i|} = 0$ or $p_j = \frac{k_j}{|W_i|} = 1$, that is $k_j = 0$ or $k_j = |W_i|$ respectively, which indicates that the decision attribute values of $W_i(i = 1, 2, \cdots, r)$ are all equal. That is to say, the discretized decision table is consistent.

Proof. – [**Theorem 2**] Let the equivalence class of the objects that have the same decision attribute value be denoted as $Y = \{Y_1, Y_2, \cdots, Y_s\}$, and the equivalence class of the objects that have identical condition attribute value be denoted as $X = \{X_1, X_2, \cdots, X_t\}$.

Since we have $R(S(P)) = 1$, then $|POC_{C^*}| = |U|$ holds. As we know $POC_{C^*}(D) \subseteq U$, then we further obtain that $POS_{C^*}(D) = U$. According to the Definition 6, for each $Y_j \in Y$, we then have at least one $X_i \in X$, to satisfy $X_i \subseteq Y_j$, and $Y_j = X_{i_1} \cup \cdots \cup X_{i_j}, (X_{i_1}, \cdots, X_{i_j} \in X)$. As it is the fact that $\bigcup X_i = \bigcup Y_j = U$, $\bigcap X_i = \bigcap Y_j = \emptyset$, then for each $X_i \in X$, there exists only one $Y_j \in Y$, so that $X_i \subseteq Y_j$. Hence, when the objects have identical condition attribute value, their decision attribute values are the same, which means the objects are consistent if $R(S(P)) = 1$.

Co-embedding of Structurally Missing Data by Locally Linear Alignment

Takehisa Yairi

Research Center for Advanced Science and Technology, University of Tokyo
yairi@space.rcast.u-tokyo.ac.jp

Abstract. This paper proposes a "co-embedding" method to embed the row and column vectors of an observation matrix data whose large portion is structurally missing into low-dimensional latent spaces simultaneously. A remarkable characteristic of this method is that the co-embedding is efficiently obtained via eigendecomposition of a matrix, unlike the conventional methods which require iterative estimation of missing values and suffer from local optima. Besides, we extend the unsupervised co-embedding method to a semi-supervised version, which is reduced to a system of linear equations. In an experimental study, we apply the proposed method to two kinds of tasks – (1) Structure from Motion (SFM) and (2) Simultaneous Localization and Mapping (SLAM).

1 Introduction

Recently, the dimensionality reduction and matrix factorization techniques have been regarded as a significant machine learning tool for feature extraction and data compression, as both the size and dimensionality of data in most application are continuing to increase rapidly.

A non-trivial issue in applying these techniques to actual problems is how to deal with *missing* data elements, as the real-world data, e.g., medical testing data, food preference questionnaire data, purchase records, etc. usually contains missing parts. If the missing portion is relatively small, *ad hoc* treatment such as filling the missing elements with constant values and inferring them from similar data is acceptable. A more sophisticated approach commonly used is to alternately estimate the missing values and conduct dimensionality reduction or matrix factorization until convergence. The method is known as EM (expectation maximization) algorithm in machine learning. However, if the missing portion is very large and has some structural pattern, these conventional approaches are expected to fail.

Consider the following situation for an example. An observer is wandering around the town, carrying a wireless device (such as tablet PC). The device is assumed to be capable of recording approximate relative directions to all detected wireless access points (APs). If the device could always communicate with all APs in the town wherever it is, the observation data could be represented as a complete matrix, whose (i, j)-th element is the relative direction to the j-th AP from the i-th observation position. Unfortunately, however, most of the elements are missing, because the wireless communication range is limited and affected by occlusion. Besides, the pattern of missing data is not random but structured, as whether a measurement is present or absent is dependent on the

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 419–430, 2012.

spatial relationship between the observer and AP. The conventional approaches are not suitable for this kind of missing data.

In this paper, we propose the locally linear alignment co-embedding (LLACoE) that embeds both row and column vectors of a matrix-form observation data with largely and structurally missing elements into low-dimensional latent spaces respectively. A key idea is that a measurement $\boldsymbol{y}_{i,j}$ can be approximated by some linear projection of the state vector of j-th object $\boldsymbol{z}_j$ onto the subspace determined by the observer's state $\boldsymbol{x}_i$. A remarkable feature of LLACoE is that it does not require iterative computation to estimate the missing values, but is efficiently solved by eigendecomposition or a system of linear equations.

2 Related Works

Dimensionality reduction is a major topic of machine learning, as well as classification, regression and clustering. Especially, in the last decade, non-linear dimensionality reduction (a.k.a. manifold learning) methods such as Isomap[7] and LLE[3] have been developed and become popular. In addition, matrix factorization or low-rank matrix approximation techniques such as singular value decomposition (SVD) and non-negative factorization (NMF) have been widely used in a variety of datamining applications.

A practical difficulty is that the real-world data is not only huge and high-dimensional, but also often incomplete due to various reasons. The simplest way of dealing with such incomplete data is to fill the missing parts with some proper constant values, typically by zero. This approach will be reasonable enough, when the values are "missing" because they are out of measurement ranges. However, the applicability of this method is obviously limited, because not all measurement data have such a property. Besides, it is sometimes nontrivial to find a proper constant value, even when it is applicable.

A more sophisticated and popular approach is to estimate the missing values and conduct dimensionality reduction or matrix factorization alternately until it converges. In computer vision (CV), PCAMD (PCA with missing data) methods[5] such as alternate least squares (ALS) and Wiberg's algorithm[1] have been utilized for the structure from motion (SFM), in which a 3-dimensional surface model of target object is estimated from a sequence of 2-dimensional images. In machine learning (ML), this kind of iterative algorithm is generally formalized as the EM algorithm. In fact, it was shown that PPCA (probabilistic PCA) with EM algorithm can deal with incomplete data[4]. Also in NMF, some iterative algorithms that alternately estimating missing values and factorizing a matrix into two low-rank ones have been recently developed[6]. While this iterative estimation approach works fine if the missing part is relatively small, the convergence property and solution quality become drastically worse as the missing portion becomes larger. Besides, even if the missing data has some pattern or structure which contains information of latent low-dimensional spaces, it does not have any mechanism to utilize the information. In summary, these conventional approaches implicitly assume small and randomly generated missing elements.

In contrast, our method utilizes only existing elements of the matrix data, which means it is not necessary to fill the absent elements with constants, nor to estimate them alternately. In addition, it takes advantage of the pattern of missing data, based on the

idea that existing (i.e. not missing) elements are roughly linear to their corresponding latent vectors.

3 Problem Definition

In this paper, we deal with a $M \times N$ data matrix $\boldsymbol{Y} = [\boldsymbol{y}_{i,j}]_{i=1,\ldots,M,j=1,\ldots,N}$. It should be noted that (i,j)-th element $\boldsymbol{y}_{i,j}$ is a D-dimensional vector in general.[1] As $\boldsymbol{Y}$ contains missing elements, we introduce a set of Boolean indicator variables $\{q_{i,j}\}$ to specify whether each element is existing or missing. That is to say,

$$q_{i,j} = \begin{cases} 0 & (\text{if}(i,j)\text{-th element } \boldsymbol{y}_{i,j} \text{ is missing}) \\ 1 & (\text{otherwise}) \end{cases} \tag{1}$$

Now we pursue two goals at the same time:

1. Obtain a set of n-dimensional row latent vectors $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M]^\top$ by reducing the dimension of $\boldsymbol{Y}$'s row vectors.
2. Obtain a set of m-dimensional column latent vectors $\boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N]^\top$ by reducing the dimension of $\boldsymbol{Y}$'s column vectors.

where, $n << N \cdot D$ and $m << M \cdot D$. It should be noted that our purpose is not to approximate or reconstruct $\boldsymbol{Y}$ by the product of $\boldsymbol{X}$ and $\boldsymbol{Z}^\top$, but to embed the row and column vectors of $\boldsymbol{Y}$ to low dimensional latent spaces respectively. It can be called as simultaneous dimensionality reduction or *co-embedding*.

We can give another view to this problem. First, assume that a measurement $\boldsymbol{y}_{i,j}$ is generated by an *unknown* function of an observer's latent state $\boldsymbol{x}_i \in \mathcal{R}^n$ and an item's latent state $\boldsymbol{z}_j \in \mathcal{R}^m$, i.e.,

$$\boldsymbol{y}_{i,j} = g(\boldsymbol{x}_i, \boldsymbol{z}_j) + \boldsymbol{e}_{i,j} \tag{2}$$

where $\boldsymbol{e}_{i,j}$ is the noise. Our goal is to estimate sets of $\{\boldsymbol{x}_i\}$ $(i = 1, \ldots, M)$ and $\{\boldsymbol{z}_j\}$ $(j = 1, \ldots, N)$, when a *partial* set of $\{\boldsymbol{y}_{i,j}\}$ is given. Note that function g itself is not necessarily estimated.

Now we make an assumption that the presence of an observation $\boldsymbol{y}_{i,j}$ has a locality as to $\boldsymbol{z}_j$. Roughly speaking, this assumption states "if there exists i such that $q_{i,j} = q_{i,j'} = 1$, then $\boldsymbol{z}_j$ and $\boldsymbol{z}_{j'}$ are close to each other".

While this assumption seems to be very restrictive, there are many problems which hold this property in fact. For example, in the case of mobile wireless device and access points mentioned in section 1, this assumption is expected to be valid because the device at a position $\boldsymbol{x}_i$ can communicate only with APs in its neighborhood. It is also the case with SLAM (simultaneous localization and mapping) problem[8] in mobile robotics, where $\boldsymbol{x}_i$ is the robot's pose and $\boldsymbol{z}_j$ is the j-th landmark's position. Another example is the SFM (structure from motion) problem in computer vision, where $\boldsymbol{x}_i$ is the relative spatial relationship between the camera and target object, $\boldsymbol{z}_j$ is the j-th visual feature's

[1] Although $\boldsymbol{Y}$ should be regarded as a $M \times N \times D$ tensor in this sense, we treat it as a matrix whose element is a vector because it makes us understand the subsequent discussion more easily.

3D coordinates in the body frame, and $\boldsymbol{y}_{i,j}$ is its 2D coordinates on the camera screen. Obviously, if j-th and j'-th features are observed at the same time, they are expected to close to each other.

The assumption may be valid even in collaborative filtering. If we consider the Netflix rating data set, $\boldsymbol{x}_i$ is the preference of the i-th user, and $\boldsymbol{z}_j$ is the j-th movie. If a user watched two movies, they are likely to be in the same genre.

4 Locally Linear Alignment Co-embedding

In this section, we introduce the proposed method named LLACoE (locally linear alignment co-embedding).

4.1 Basic Idea

We consider the above assumption "if there exists i such that $q_{i,j} = q_{i,j'} = 1$, then $\boldsymbol{z}_j$ and $\boldsymbol{z}_{j'}$ are close to each other" holds. Then, if $q_{i,j} = 1$ or $\boldsymbol{y}_{i,j}$ is not missing, a linear approximation below is possible in its neighborhood, i.e.,

$$\boldsymbol{y}_{i,j} = g(\boldsymbol{x}_i, \boldsymbol{z}_j) + \boldsymbol{e}_{i,j} \approx \boldsymbol{G}(\boldsymbol{x}_i)[\boldsymbol{z}_j^\top, 1]^\top = \boldsymbol{G}(\boldsymbol{x}_i)\tilde{\boldsymbol{z}}_j \tag{3}$$

where $\boldsymbol{G}(\boldsymbol{x}_i)$ stands for a projection matrix determined by $\boldsymbol{x}_i$, and $\tilde{\boldsymbol{z}}_j$ is a homogeneous coordinates of $\boldsymbol{z}_j$.

Assume that $\boldsymbol{x}_i$ implies observer's latent state at time i, while $\boldsymbol{z}_j$ implies j-th object's state or position. Then the above approximation states that when observer's state is $\boldsymbol{x}_i$, its observation data is formed by linear projections of all observable objects $j \in \mathcal{V}_i$ into the observation subspace $\boldsymbol{G}(\boldsymbol{x}_i)$ determined by $\boldsymbol{x}_i$. In other words, each observation data at a time can be regarded as linear projections of a piece (fragment) of the whole world's state into a low-dimensional perception space.

Now, our first goal is to reconstruct the latent states of all objects, i.e., $\{\boldsymbol{z}_j\}$ by *aligning* the pieces of observation data. Intuitively, it is similar to jigsaw puzzles or reconstruction of fragmentary fossils. Since the alignment operation of each piece reflects the observer's state, $\boldsymbol{x}_i$ is also expected to be reconstructed. In the remaining of this section, we will explain how to realize this rough idea.

4.2 Unsupervised Locally Linear Alignment Co-embedding

First we consider reconstructing the column latent vectors $\{\boldsymbol{z}_j\}$. The assumption in the previous section means that $\boldsymbol{z}_j$ is approximately linear (more strictly, affine) to $\boldsymbol{y}_{i,j}$ if $q_{i,j} = 1$. We use this local linearity property in a reverse way. That is to say, we think of approximating $\boldsymbol{z}_j$ by an affine transformation of $\boldsymbol{y}_{i,j}$ when $q_{i,j} = 1$:

$$\hat{\boldsymbol{z}}_{i,j} \equiv \boldsymbol{T}_i[\boldsymbol{y}_{i,j}^\top, 1]^\top = \boldsymbol{T}_i\tilde{\boldsymbol{y}}_{i,j} \tag{4}$$

where $\boldsymbol{T}_i$ is an alignment transformation matrix common for $\boldsymbol{y}_{i,j}$ $(j = 1, \ldots, N)$ as long as $q_{i,j} = 1$. $\tilde{\boldsymbol{y}}_{i,j}$ is the homogeneous coordinates of $\boldsymbol{y}_{i,j}$. It would be reasonable to decide the final estimate of $\boldsymbol{z}_j$ by averaging all the temporary estimates as:

$$\hat{\boldsymbol{z}}_j = \frac{\sum_{i=1}^{M} q_{i,j}\hat{\boldsymbol{z}}_{i,j}}{\sum_{i=1}^{M} q_{i,j}} = \sum_{i=1}^{M} \tilde{q}_{i,j}\hat{\boldsymbol{z}}_{i,j} \tag{5}$$

where, $\tilde{q}_{i,j} = q_{i,j}/\sum_{i=1}^{M} q_{i,j}$ is the normalized observability indicator.

Now our main concern is how we can obtain the optimal set of alignment matrices $\{\boldsymbol{T}_i\}$ $(i = 1, \ldots, M)$. A reasonable way is to choose them so that $\{\hat{\boldsymbol{z}}_{i,j}\}$ $(i = 1, \ldots, M)$– the estimates of $\boldsymbol{z}_j$ for all i coincide with each other. This idea can be realized by minimizing the following cost function Φ_{aln} with respect to $\{\boldsymbol{T}_i\}$:

$$\Phi_{aln} = \frac{1}{2}\sum_{j=1}^{N}\sum_{i\neq i'} \tilde{q}_{i,j}\tilde{q}_{i',j}\|\hat{\boldsymbol{z}}_{i,j} - \hat{\boldsymbol{z}}_{i',j}\|^2 \tag{6}$$

Although we omit the detailed derivation here, by introducing some auxiliary matrices and vectors such as $\boldsymbol{v}_j = [\tilde{q}_{1,j}\tilde{\boldsymbol{y}}_{1,j}^\top, \ldots, \tilde{q}_{M,j}\tilde{\boldsymbol{y}}_{M,j}^\top]$, $\boldsymbol{V} = [\boldsymbol{v}_1^\top, \ldots, \boldsymbol{v}_N^\top]^\top$, $\boldsymbol{D}_i = \sum_{j=1}^{N} \tilde{q}_{i,j}\tilde{\boldsymbol{y}}_{i,j}\tilde{\boldsymbol{y}}_{i,j}^\top$, $\boldsymbol{D} = diag(\boldsymbol{D}_1, \ldots, \boldsymbol{D}_M)$, $\boldsymbol{T} = [\boldsymbol{T}_1, \ldots, \boldsymbol{T}_M]^\top$, Eq.(6) can be rewritten as:

$$\Phi_{aln}(\boldsymbol{T}) = Tr(\boldsymbol{T}^\top(\boldsymbol{D} - \boldsymbol{V}^\top\boldsymbol{V})\boldsymbol{T}) \tag{7}$$

Note that this is a trace of a matrix quadratic form of $\boldsymbol{T}$, and that $\boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N]^\top$ can be obtained as $\boldsymbol{Z} = \boldsymbol{V}\boldsymbol{T}$.

As the minimization of Φ_{aln} has a trivial solution $\boldsymbol{T} = \boldsymbol{0}$ if there are no constraints, we impose a constraint :

$$\boldsymbol{Z}^\top\boldsymbol{Z} = \boldsymbol{T}^\top(\boldsymbol{V}^\top\boldsymbol{V})\boldsymbol{T} = \boldsymbol{I} \tag{8}$$

The solution of this constrained minimization is obtained as $\boldsymbol{T}_{opt} = [\boldsymbol{u}_2, \ldots, \boldsymbol{u}_{m+1}]$ where $\boldsymbol{u}_2$, $\boldsymbol{u}_{m+1}$ are the second smallest and $(m+1)$-smallest eigenvectors of the generalized eigenvalue problem:

$$(\boldsymbol{D} - \boldsymbol{V}^\top\boldsymbol{V})\boldsymbol{u} = \lambda(\boldsymbol{V}^\top\boldsymbol{V})\boldsymbol{u} \tag{9}$$

Then we obtain $\hat{\boldsymbol{Z}} = \boldsymbol{V}\boldsymbol{T}_{opt}$.

Next we consider reconstructing the row latent vectors $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M]^\top$. As each alignment transformation matrix $\boldsymbol{T}_i$ obtained in the previous step is supposed to characterize the corresponding row latent vector, estimates of $\{\boldsymbol{x}_i\}$ are obtained by reducing the dimension of $vec(\boldsymbol{T}_i)$ to n, where $vec(\boldsymbol{T}_i)$ is a column vector obtained by reshaping the elements of matrix $\boldsymbol{T}_i$. Note that $\{vec(\boldsymbol{T}_i)\}$ contain no missing elements, unlike the original observation matrix $\boldsymbol{Y}$. We employed the simple SVD for the dimensionality reduction this time, while other advanced non-linear methods are also applicable.

The above cost function and the solution of column latent vectors $\{\boldsymbol{z}_j\}$ originate from Verbeek and Roweis's method for non-linear PCA and CCA[9]. However, they did not deal with the missing elements nor simultaneous dimensionality reduction of column and row vectors. Therefore, our method is different from theirs.

4.3 Regularization

In actual applications, we can often improve the estimation results by introducing task-specific regularization terms into the original cost function. Especially, when the row latent vector $\boldsymbol{x}_i$ corresponds to the observer's state at time i, each pair of $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+1}$ and corresponding pair of alignment matrices $\boldsymbol{T}_i$ and $\boldsymbol{T}_{i+1}$ are expected to be close to each other. This soft constraint can be realized by introducing a regularization term for smoothing successive rows of $\boldsymbol{X}$ expressed as,

$$\Phi_{smo} = Tr(\boldsymbol{T}^\top(\boldsymbol{S}^\top\boldsymbol{S})\boldsymbol{T}) \tag{10}$$

where $\boldsymbol{S}$ is a matrix that computes the differences of pairs of successive elements in $\boldsymbol{T}$. We minimize the weighted sum of cost functions $\Phi_{aln} + \alpha_{smo} \cdot \Phi_{smo}$ instead of Φ_{aln} under the same constraint.

4.4 Semi-supervised Co-embedding

In some application domains, a semi-supervised problem setting where the partial label information about row and column latent vectors are available beforehand is more natural. For example, in the case of wireless device and access points story, it is no wonder that exact positions of observer are partially available by GPS. LLACoE can be extended to a semi-supervised version in a straightforward way.

We denote the labeled data of j-th column latent vector $\boldsymbol{z}_j$ as $\boldsymbol{z}_j^*$. We also define a Boolean variable δ_j to indicate whether the label information is available or not. That is to say,

$$\boldsymbol{z}_j^* = \boldsymbol{z}_j \ (\text{if } \delta_j = 1), \qquad \boldsymbol{0} \ (\text{if } \delta_j = 0) \tag{11}$$

Then we define the cost function for the label information as:

$$\Phi_{zlb} \equiv \sum_{j=1}^{N} \delta_j \|\hat{\boldsymbol{z}}_j - \boldsymbol{z}_j^*\|^2 \tag{12}$$

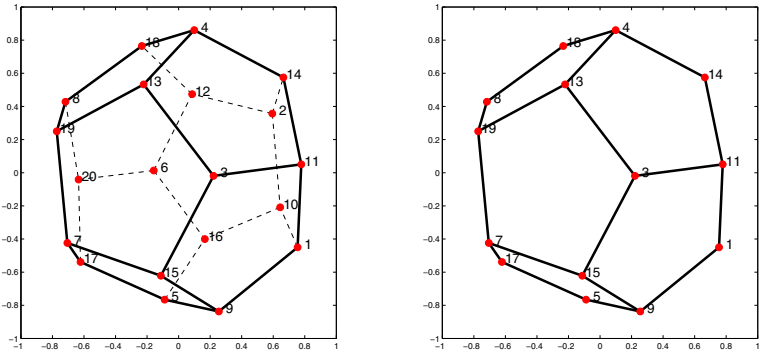
By defining $\boldsymbol{Z}^* = [\boldsymbol{z}_1^*, \ldots, \boldsymbol{z}_N^*]^\top$ and $\boldsymbol{J}_z = diag(\delta_1, \ldots, \delta_N)$, Eq.12 can be re-written as,

$$\Phi_{zlb} = Tr((\boldsymbol{V}\boldsymbol{T} - \boldsymbol{Z}^*)^\top \boldsymbol{J}_z(\boldsymbol{V}\boldsymbol{T} - \boldsymbol{Z}^*)) \tag{13}$$

The whole cost function $\Phi_{sem}(\boldsymbol{T}) = \Phi_{aln} + \alpha_{smo} \cdot \Phi_{smo} + \alpha_{zlb} \cdot \Phi_{zlb}$ can be easily minimized by solving a system of linear equations:

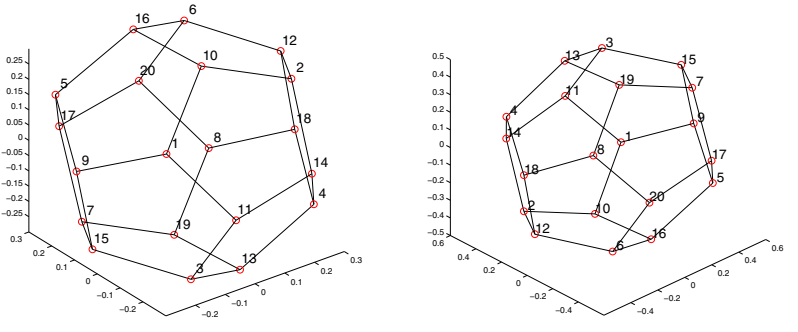
$$\boldsymbol{T}_{opt} = (\boldsymbol{D} + \boldsymbol{V}^\top(\alpha_{zlb}\boldsymbol{J}_z - \boldsymbol{I})\boldsymbol{V} + \alpha_{smo}\boldsymbol{S}^\top\boldsymbol{S})^{-1}(\alpha_{zlb}\boldsymbol{V}^\top\boldsymbol{J}_z\boldsymbol{Z}^*) \tag{14}$$

Introducing the label information of row latent vectors $\{\boldsymbol{x}_i^*\}$ is similar to the above discussion, but much simpler. It is a general semi-supervised regression problem, where $\{vec(\hat{\boldsymbol{T}}_i)\}$ are input vectors. While there are many advanced methods for the semi-supervised regression, this time we solved it simply by the ridge regression or least-squares linear regression with Tiknov regularization.



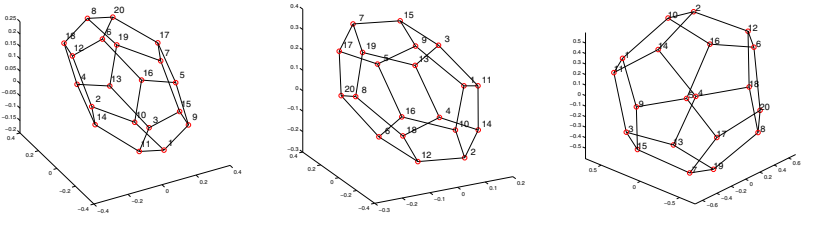
(a) Complete Observation (b) Incomplete Observation

Fig. 1. Examples of (a) complete and (b) incomplete observation in Exp.1. Numbers indicate vertices's' IDs



(a) SVD (b) (Unsupervised) LLACoE

Fig. 2. Reconstructed 3D model of dodecahedron with complete observation data



(a) ALS (18.9 sec) (b) Wiberg (16.4 sec) (c) LLACoE (0.17 sec)

Fig. 3. Reconstructed 3D model of dodecahedron with incomplete observation data with computational time. All algorithms are implemented in Matlab and conducted by a Dell Precision T1500.

5 Experiment

5.1 Experiment 1: Structure from Motion Task

First, we applied the proposed co-embedding method to the structure from motion (SFM) task in computer vision domain, and compared it with conventional methods.

Assume that we look at a dodecahedron from a randomly chosen direction, identify all visible vertices, then obtain their 2-dimensional coordinates on camera image as the observation data $[\boldsymbol{y}_{i,1}, \ldots, \boldsymbol{y}_{i,N}]$, where $N = 20$ because a dodecahedron has 20 vertices. We repeat this procedure for $M = 100$ times, and obtain the observation data $\boldsymbol{Y}$. The goal of this task is to reconstruct a 3-dimensional model of dodecahedron, or estimate 3-D coordinates $\{\boldsymbol{z}_j\}$ of 20 vertices in the body frame.

For comparison, we first conducted this experiment under the condition that all vertices are always visible, i.e., $\boldsymbol{Y}$ has no missing elements (Fig.1 (a)). In this case, ordinary SVD is applicable. In fact, a perfect 3-D model is reconstructed by SVD as Fig.2 (a). Unsupervised version of the proposed method (LLACoE) also succeeds in reconstructing it as Fig.2 (b).

Next we impose the practical condition that observation elements of occluded vertices are lost (Fig.1 (b)). As a result, approx. 30 % of $\boldsymbol{Y}$'s elements are missing. In this case, we cannot use the ordinary SVD anymore, because filling the missing elements with some constants is obviously inappropriate. So we applied two PCAMD methods, i.e., alternate least squares (ALS) algorithm and Wiberg's algorithm[1]. The resultant models are shown in Fig.3 (a)-(c). Although all three methods reconstructed the model successfully, LLACoE is much faster than others because it does not need iterations.

5.2 Experiment 2: Mapping and Localization for Wireless Devices

Next we applied LLCoE to a simultaneous localization and mapping (SLAM) problem with wireless devices in a simulated environment.

In this task, we assume that 564 access points (APs) are distributed in a virtual campus, and a walking observer with a wireless client device records the relative positions of detected APs periodically. Fig.4 illustrates the simulated environment (research campus) and the ground truth map of APs. Some APs' IDs are indicated for later evaluation. Fig.5 illustrates the ground truth trajectory of the observer and observation points. Number of observation points is 310. In this task, the row latent vector $\boldsymbol{x}_i$ $(i = 1, \ldots, 310)$ is the observer's state (i.e., position and heading direction), whereas the column latent vector $\boldsymbol{z}_j$ $(j = 1, \ldots, 564)$ is each AP's position. Observation data $\boldsymbol{y}_i, j$ is computed from a very noisy bearing and range information. For example, Fig.6 (a) and (b) are a ground truth map and a observed relative positions of detected APs at one time. We generated the observation data with:

$$Pr(q_{i,j} = 1) = \frac{1}{1 + \exp(0.15 \cdot (d_{i,j} - 50))} \tag{15}$$

where $d_{i,j}$ is the distance between i-th observation point and j-th AP. As a result, the ratio of missing elements in $\boldsymbol{Y}$ becomes approx. 97 %. Fig.7 shows the distribution of missing (gray) and existing (white) elements in $\boldsymbol{Y}$.

Fig. 4. Simulated environment with 564 AP positions

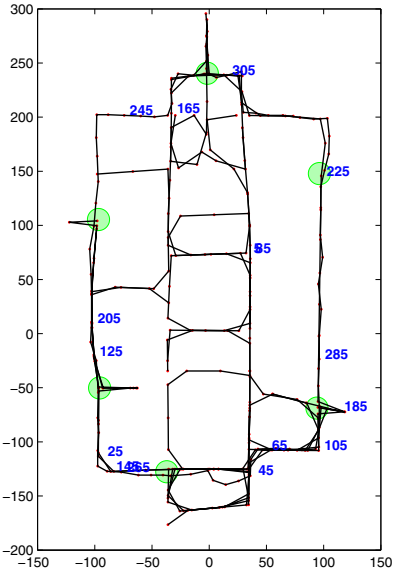
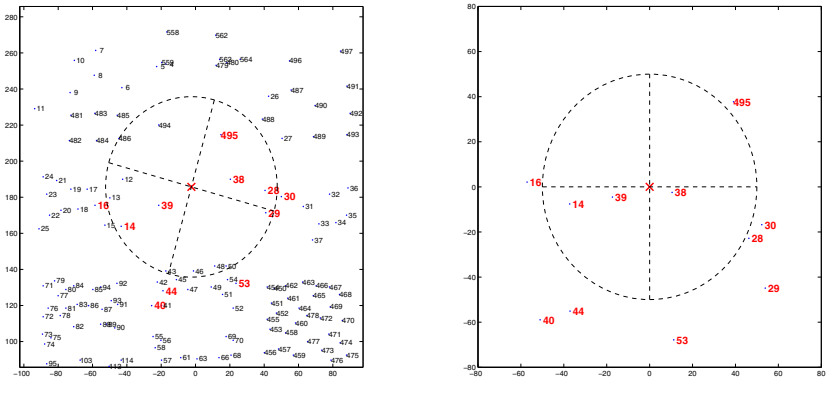


Fig. 5. Ground truth trajectory of observer



(a) Ground truth (b) Observed

Fig. 6. Example of ground truth submap (a) and observed data (b). Observation is noisy and distant APs are missing. Circles show the approximate communicatable ranges.

Unsupervised Localization and Mapping. First we applied the unsupervised version of LLACoE to estimate $\boldsymbol{X}$ and $\boldsymbol{Z}$ from $\boldsymbol{Y}$ without the smoothing regularization. Although the map of APs ($\boldsymbol{Z}$) in Fig. 8(a) is largely distorted, we can see the approximate relative relationships with neighbors are reconstructed to some extent. On the other hand, the trajectory of observer ($\boldsymbol{X}$) in Fig. 9(a) is reconstructed very well.

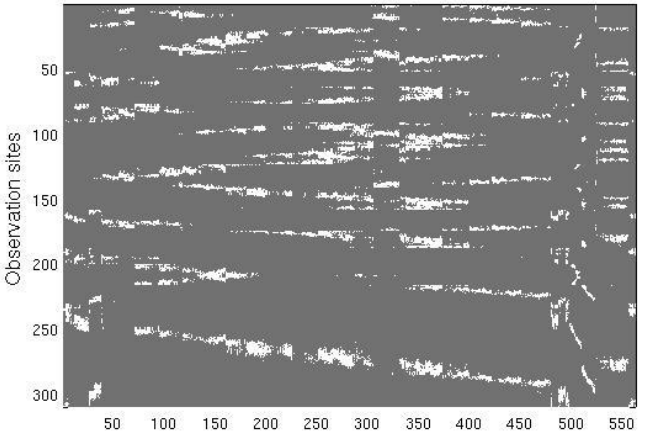
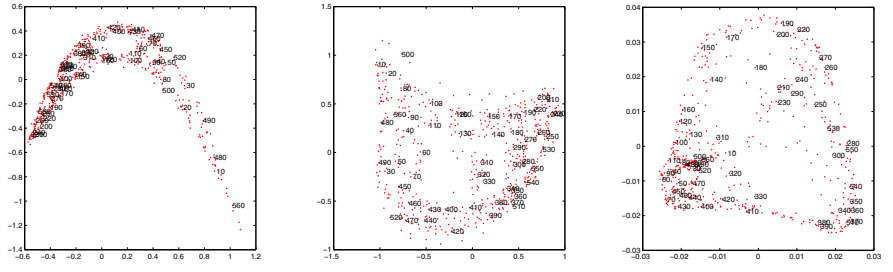
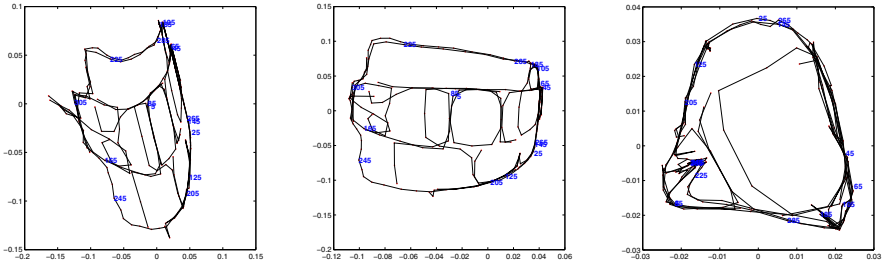


Fig. 7. Distribution of missing (gray) and existing (white) elements of observation data $\boldsymbol{Y}$. About 97% is missing.



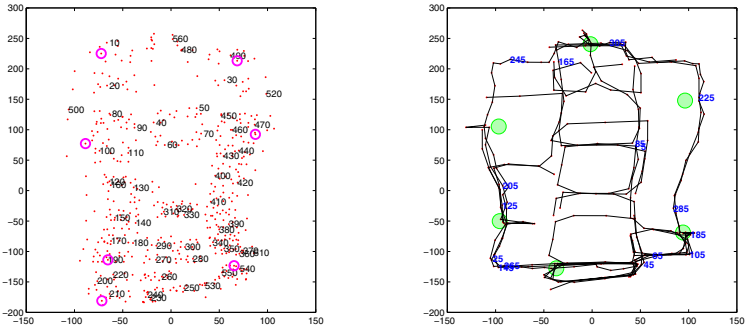
(a) LLACoE w/o smoothing (b) LLACoE with smoothing (c) LSI

Fig. 8. Reconstructed maps by unsupervised methods



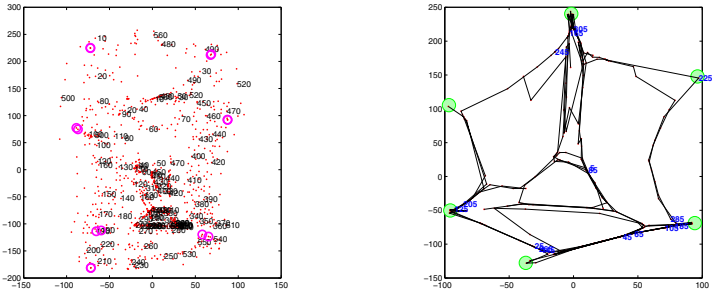
(a) LLACoE w/o smoothing (b) LLACoE with smoothing (c) LSI

Fig. 9. Reconstructed trajectories by unsupervised methods



(a) Map (b) Trajectory

Fig. 10. Estimated map and trajectory by semi-supervised LLACoE



(a) Map (b) Trajectory

Fig. 11. Estimated map and trajectory by co-localization[2]

Then we added the smoothing regularization term Φ_{smo} described in section 4.3. We set the weight parameter value as $\alpha_{smo} = 0.2$ here. The results are shown in Fig.8(b) and Fig.9(b). We can see that the reconstruction of $\boldsymbol{X}$ (map of APs) is much improved.

For comparison, we applied latent semantic indexing (LSI) method as in [2] to the data. To do so, we converted the range measurements into signal strengths by a monotonically decreasing function. The results are much worse than those of LLACoE as shown in Fig.8(c) and Fig.9(c).

Semi-supervised Localization and Mapping. We also tested the semi-supervised version of LLACoE in this experiment. We gave exact positions of 7 APs as the label information $\boldsymbol{z}_j^*$, which are emphasized by circles in Fig.4. Partial label information of observation points $\boldsymbol{x}_i$ were also provided within the "areas" indicated in Fig.5.

Fig.10 (a) and (b) show the obtained map and trajectory, respectively. Owing to the label information, the absolute accuracy of estimated positions is much improved.

For comparison, we applied Pan's co-localization algorithm based on graph regularization [2] to the range measurements. The resultant map and trajectory are shown in Fig.11 (a) and (b). Unfortunately, it completely failed in this experiment.

6 Conclusion

In this paper, we proposed a co-embedding method to embed the row and column vectors of an observation matrix data whose large portion is structurally missing into low-dimensional latent spaces simultaneously. The proposed method outperforms the conventional methods based on EM algorithm and ALS in computational cost and stability, because it is solved by eigendecomposition of a symmetric matrix. We also a semi-supervised version of the proposed co-embedding method, which is solved by a system of linear equations. In the experiment, we evaluated the method on two kinds of tasks, and compared it with other methods. In future, we are going to apply this method to a variety of problems.

References

1. Okatani, T., Deguchi, K.: On the wiberg algorithm for matrix factorization in the presence of missing components. International Journal of Computer Vision 72(3), 329–337 (2007)
2. Pan, J., Yang, Q.: Co-localization from labeled and unlabeled data using graph laplacian. In: Proceedings of IJCAI 2007, pp. 2166–2171 (2007)
3. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
4. Roweis, S.: Em algorithms for pca and spca. In: in Advances in Neural Information Processing Systems, pp. 626–632 (1998)
5. Shum, H.Y., Ikeuchi, K., Reddy, R.: Principal component analysis with missing data and its application to polyhedral object modeling. IEEE Trans. Pattern Anal. Mach. Intell. 17(9), 854–867 (1995)
6. Sindhwani, V., Bucak, S.S., Hu, J., Mojsilovic, A.: One-class matrix completion with low-density factorizations. In: Proceedings of the 2010 IEEE International Conference on Data Mining, pp. 1055–1060 (2010)
7. Tenenbaum, J.B., Silva, V.D., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
8. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press (2005)
9. Verbeek, J., Roweis, S.T., Vlassis, N.: Non-linear cca and pca by alignment of local models. In: Procs. of NIPS (2004)

Relevant Feature Selection from EEG Signal for Mental Task Classification

Akshansh Gupta and R.K. Agrawal

School of Computer and Systems Sciences
Jawaharlal Nehru University, New Delhi India 1110067
{akshansgupta83,rkajnu}@gmail.com

Abstract. In last few years, the research community has shown interest in the development of Brain Computer Interface which may assists physically challenged people to communicate with the help of brain signal. The two important components of such BCI system are to determine appropriate features and classification method to achieve better performance. In literature, Empirical Mode Decomposition is suggested for feature extraction from EEG which is suitable for the analysis of non-linear and non-stationary time series. However, the features obtained from EEG may contain irrelevant and redundant features which make them inefficient for machine learning. Relevant features not only decrease the processing time to train a classifier but also provide better generalization. Hence, relevant features which provide maximum classification accuracy are selected using ratio of scatter matrices, Chernoff distance measure and linear regression. The performance of different mental task using different measures used for feature selection is compared and evaluated in terms of classification accuracy. Experimental results show that there is significant improvement in classification accuracy with features selected using all feature selection methods and in particular with ratio of scatter matrices.

Keywords: Empirical Mode Decomposition, Brain Computer Interface, Feature Selection, Chernoff distance measure, Scatter Matrices, Linear regression.

1 Introduction

Last few years have witnessed the advancement of technologies which has made possible the use of brain signals for communication between human and computer. This growth in technologies allows research community to develop a system called Brain Computer Interface (BCI) which can control a device such as computer or wheel chair by human intentions rather than mechanical power of human. It may be very useful to physically challenged persons who are suffering from locomotor syndrome, Amyotrophic Lateral Sclerosis, Head trauma, severe cerebral palsy or multiple disorders affect in body, which restricts such persons to operate any electronics device smoothly and freely. With the development of BCI, these people can operate any electronics device with the help of just

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 431–442, 2012.

brain signals and does not depend on the brain's normal output pathways of peripheral nerves and muscles. BCIs are often aimed for assisting, augmenting or repairing human cognitive or motor sensory function. Various techniques such as Electroencephalogram (EEG), Electrocardiogram, functional magnetic resonance imaging, Magneto encephalographic (MEG) and Positron emission tomography (PET) are used for monitoring brain signals activities.

EEG is commonly used for BCI implementation due to its low cost, ability to record brain signals and non-invasive nature. There are many components of a BCI system. However, the success of BCI system mainly depends on two components: feature extraction and classification method. The feature extracted/selected from EEG should have high discriminative power to distinguish the different tasks and the classification methods used to distinguish the different tasks should be efficient in real time. There are many classification methods available in the field of data mining and machine learning[16,20,31]. The research work[22] discusses pros and cons of linear and classification methods for BCI research.

In literature, autoregressive (AR) models or adaptive AR models (AAR)[1,3,7,9,20,26] and power spectral density (PSD)[2,27] are commonly used for feature extraction from EEG for BCI system. However, these methods assume linearity, Gaussianality and minimum-phase within EEG signals, i.e., the amplitudes of EEG signals are normally distributed, their statistical properties do not vary over time, and their frequency components are uncorrelated. Under these assumptions, the EEG signal is considered as a linear superposition of statistically independent sinusoidal or other wave components, and only frequency and power estimates are considered while phase information is lost. Recently, Empirical Mode Decomposition (EMD) is suggested for feature extraction from EEG signal which is suitable for the analysis of non-linear and non-stationary time series. A disadvantage arising at this point is that the feature vector so obtained with EMD would be too large and the number of training samples available are in general relatively small number. Consequently, it is essential to do a feature selection in order to solve the problem of curse-of-dimensionality which arises due to small sample and large number of features[17]. Also, the resultant features may contain noisy, irrelevant or redundant features which make them inefficient for machine learning. In fact, the presence of irrelevant and redundant features may deteriorate the performance of the classifier and requires high computation time and other resources for training and testing the data. Hence, in order to enhance the performance of BCI system in terms of accuracy and time required to detect, there is need to identify a set of relevant features.

Feature selection is used to remove such noisy, irrelevant, and redundant features.There are two major approaches to feature selection: filter and wrapper approach[10,14,22]. Most filter methods employ statistical characteristics of data for feature selection which requires less computation. It independently measures the importance of features without involving any classifier. Since, the filter approach does not take into account the learning bias introduced by the final learning algorithm, it may not be able to select the most relevant set of features

for the learning algorithm. On the other hand, wrapper methods tend to find features better suited to the predetermined learning algorithm resulting in better performance. But, it tends to be computationally more expensive since the classifier must be trained for each candidate subset.

Feature ranking approaches have been widely investigated for feature selection[10,21,23] in literature. Since in most of feature ranking approaches, features are evaluated using statistical characteristics of the data, different feature ranking methods measure different characteristics of data. Therefore, the informative features selected by different ranking methods may be different. In literature to remove redundancy a forward/backward feature selection method or its combinations are used with a measure that selects relevant and non redundant features. Among the most widely used filter methods for feature selection, there are techniques based on statistical separability measures which allow one to select a suitable subset of features by assigning the degree of interclass separability associated with each subset. In particular, ratio of scatter matrices, Chernoff distance measures[19] and linear regression[21] are commonly employed by research community in various area of data mining and pattern classification field but yet to be explored in feature selection of EEG data for mental task classification. In this paper, we compare and evaluate these measures to determine relevant features for BCI system.

Our work is organized as follows: Feature extraction using empirical mode decomposition is included in Sect. 2. A brief introduction of separability measures employed for features selection are discussed in Sect. 3. Experimental data and results are discussed in Sect. 4 and Sect. 5 contains conclusions.

2 Feature Extraction from EEG

The feature extraction is carried out in two phases [12]: in the first phase, the empirical mode decomposition is used, and the second phase estimates different time and frequency parameters.

2.1 Empirical Mode Decomposition (EMD)

Under the assumption that any signal is composed of a series of different intrinsic oscillation modes, the EMD can be used to decompose the incoming signal into its different Intrinsic Mode Function (IMF). An IMF is a function that satisfies two conditions[12]:

1. In the entire signal, the number of extremes and the zero-crossings must be equal or differ at most by one.
2. At any point, the mean value of the envelope defined by the local maxima and the envelope defined by the local minima must be zero.

Given the incoming signal x(t), the algorithm of EMD is based on a sifting process that can be summarized as[12,14]:

1. Interpolate all the local maxima and minima in the signal with a cubic spline line, to produce the upper and lower envelope.
2. Repeat for the local minima to produce the lower envelope.
3. Compute the mean of both envelopes m_1.
4. Extract the detail h_1=$x(t) - m_1$
5. Repeat the steps 1 to 4, and consider the detail h_i as the data, until detail h_1 can be considered an IMF.
6. After k iterations, the detail h_k is an IMF and is designated as: $IMF_1 = h_k$
7. Iterate steps 1 to 6 on the residual r_j in order to obtain all the IMFs of the signal:

$$r_j = x\,(t) - IMF_1 - IMF_2 ... IMF_m \tag{1}$$

The procedure terminates when the residual r_j is either a constant, a monotonic slope, or a function with only one extreme. The result of the EMD process produces n IMFs and a residue signal r_n. The original signal x(t) can be reconstructed summing up the n extracted IMF and the residue:

$$x(t) = \sum_{j=1}^{n} IMF_j + r_j \tag{2}$$

2.2 Estimation of Various Parameters

In order to obtain the IMFs of the signal, publicly available EMD toolbox for Matlab was utilized. The lower-order IMFs capture the faster oscillation modes of the signal, whereas the higher-order IMFs capture the slower oscillation modes. The EMD algorithm can be applied to each EEG 1 s segments. Afterward, the EMD is able to extract no more than five IMFs and the residue for each 1 s EEG segment. For each one of these five IMFs, different parameters can be computed. The following parameters can be used to represent each EMD[5]:

1. Root Mean Square (RMS),
2. Variance,
3. Shannon entropy [23]
4. Lempel-Ziv Complexity Measure [13],
5. Central Frequency (50 % of spectrum energy)
6. Maximum Frequency (95 % of spectrum energy)

Some parameters were chosen since they are commonly used in BCI (RMS, variance), LZ quantifies the complexity of a signal analysing its spatial-temporal patterns and was used to analyse EEG signals in other areas [10]. The central and maximum frequencies were used as descriptors of the bandwidth of each IMF. Entropy was used to measure the average amount of information in a signal.

3 Feature Selection

A disadvantage arising at this point is feature vector contains 180 parameters (5 IMFs x 6 parameters 6 channels). Consequently, it is essential to do a feature

selection in order to solve the curse-of-dimensionality inconvenience[24]. Feature ranking is commonly used to determine a subset of relevant features. However, the disadvantage of feature ranking method is that they ignore the correlations between features. Hence the features selected may contain redundant information and influences the classification capabilities of the feature subset that is selected. Some of the methods suggested in literature for removing redundancy are Chernoff distance measure[22], ratio of inter-class and with-in class scatter, and linear regression[19]. In order to obtain a quantitative measure of how separable are two classes, a distance measure can be easily extracted from some parameters of the data. A very important aspect of probabilistic distance measures is that a number of these criteria can be analytically simplified in the case when the class conditional p.d.f.s follows multivariate normal distribution. The class conditional probability densities functions $p(X_k|C_i)$ of k-dimensional samples $\mathbf{X_k} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_k]$ for a given class $C_i, i = 1, 2, 3, ..., k$ is given by

$$p(\mathbf{X}_k|C_i) = \frac{1}{(2\pi)^{\frac{d}{2}}\left|\sum_{\mathbf{k}}^{\mathbf{i}}\right|} \exp\left[-\frac{1}{2}(\mathbf{X}_k - \mu_k^i)^T(\sum_k^i)^{-1}(\mathbf{X}_k - \mu_{\mathbf{k}})\right] \tag{3}$$

where $\mu_{\mathbf{k}}^{\mathbf{i}}$ is a mean vector and $\sum_k^i$ is a covariance matrix for class C_i . In literature, for multivariate normal distribution for two classes, CD measure is given as follows[5]:

$$J_k^c = \frac{1}{2}\boldsymbol{\beta}(1-\boldsymbol{\beta})(\mu_k^2 - \mu_k^1)^T[(1-\boldsymbol{\beta})\sum_k^1 + \boldsymbol{\beta}\sum_k^2]^{-1}(\mu_k^2 - \mu_k^1) + \frac{1}{2}log\frac{\left|(1-\boldsymbol{\beta})\sum_k^1 + \boldsymbol{\beta}\sum_k^2\right|}{\left|\sum_k^1\right|^{1-\boldsymbol{\beta}}\left|\sum_k^2\right|^{\boldsymbol{\beta}}} \tag{4}$$

A major disadvantage of the class separability measure CD is that it is not easily computed, unless the Gaussian assumption is employed. In literature, a simpler criteria based on the scatters of feature vector samples is employed. To this end, the scatter matrices: within-class scatter and between-class scatter are respectively defined as:

$$S_w = \sum_k^1 + \sum_k^2 \tag{5}$$

$$S_b = (\mu_k^2 - \mu_k^1)(\mu_k^2 - \mu_k^1) \tag{6}$$

From these definition of scatter matrices, it is straightforward to observe that the criterion

$$J = \frac{|S_b|}{|S_w|} \tag{7}$$

takes large values when samples of the selected features space are well clustered around their mean within each class, and the clusters of the different classes are well separated. Also, the criteria J have the advantage of being invariant under linear transformation.

The regression analysis considers the relations between the selected features which minimizes redundancy. While using regression analysis for data, a multiple

regression model is considered because there can be many features which could affect the presence or absence of samples from a particular class. A multiple regression model with a target variable y and multiple variables X is given by [15]:

$$y = \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1 x_{i1} + \boldsymbol{\beta}_2 x_{i2} + ... + \boldsymbol{\beta}_m x_m + \varepsilon_i, i = 1, 2, ...m \tag{8}$$

Where $\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, ...\boldsymbol{\beta}_m$ are constants estimated by observed values of X and class label y and is estimated by normal distribution having mean zero and a variance σ^2. The sum of square errors (SSE) is given by

$$SSE = \sum_{i=0}^{n} (y_i - y_i^p) \tag{9}$$

Where y and $y^p$are observed and predicted values respectively. A large value of SSE means that the regression is predicted poorly. The total sum of squares is given by

$$SSTO = \sum_{i=0}^{n} (y_i - \bar{y}) \tag{10}$$

Where $\bar{y}$ is the average of y_i. In a regression model the choice of features which best explains the class label depends on the value of R^2 which is given by

$$R^2 = 1 - \frac{SSE}{SSTO} \tag{11}$$

4 Experimental Set-Up and Results

The EEG data used in our experiment was acquired by Keirn and Aunon [29] using the following procedure. The subjects were seated in an Industrial Acoustics Company sound controlled booth with dim lighting and noiseless fans for ventilation. An Electro-Cap elastic electrode cap was used to record from positions C3, C4, P3, P4, O1, and O2, defined by the 10-20 system of electrode placement. The electrodes were connected through a bank of Grass 7P511 amplifiers and bandpass filtered from 0.1100Hz. Data was recorded at a sampling rate of 250 Hz with a Lab Master 12 bit A/D converter mounted in an IBM-AT computer. Eye blinks were detected by means of a separate channel of data recorded from two electrodes placed above and below the subjects left eye.

For our experiment, the data from six subjects except subject 5 performing five different mental tasks were analyzed. The five mental tasks are: the baseline(B) task, for which the subjects were asked to relax as much as possible; the letter(L) task, for which the subjects were instructed to mentally compose a letter to a friend or relative without vocalizing; the math(M) task, for which the subjects were given non-trivial multiplication problems, such as 49 times 78, and were asked to solve them without vocalizing or making any other physical movements; the visual counting(C) task, for which the subjects were asked to imagine a blackboard and to visualize numbers being written on the board sequentially; and

the geometric figure rotation(R), for which the subjects were asked to visualize a particular three-dimensional block figure being rotated about an axis.

Data was recorded for 10 seconds during each task and each task was repeated five times per session. Most subjects attended two such sessions recorded on separate weeks, resulting in a total of 10 trials for each task. With a 250 Hz sampling rate, each 10 second trial produces 2,500 samples per channel. These are divided into half-second segments that overlap by one quarter-second, producing at most 39 segments per trial segments containing eye blinks are discarded.

Features are extracted from each one of signal using EMD. Each signal is represented in terms of 180 statistics (5 IMFs x 6 parameters 6 channels). To remove redundancy from the selected pool of features, three feature selection measures are investigated: Chernoff distance measure, ratio of with-in class scatter and between class scatter and linear regression. For Chernoff distance measure, features are selected using 3 different values ranging from 0.1 to 0.9 with an increment of 0.4. We have used linear discriminate classifier (LDC), Quadratic discriminate classifier (QDC), k-nearest neighbor (KNNC) and Support vector machine (SVC) to evaluate the performance of the feature selection methods. The average classification accuracy is computed using ten cross-validations. All the simulations are done using matlab. Tables 1 and Table 2 show the minimum classification accuracy achieved with different classifiers and the number of features for different measures respectively. For Chernoff distance measure, the maximum classification accuracy achieved over different values of β is shown in Table 1. The best results in each category are indicated in bold. Figures 1-2 and Tables 1-2 show the variation of classification accuracies and minimum number of features for the different mental tasks respectively. Figures 1-2 shows at end of this manuscript. We observe the following from Tables 1-2:

1. The classification accuracy of all mental tasks classification improved significantly with the use of feature selection.
2. The maximum average classification accuracy of mental tasks is achieved with feature selection method using ratio of scatter matrices for all classifiers except KNN.
3. The average classification accuracy of mental tasks with SVC and LDC are similar and better in comparison to QDC and KNN using all feature selection methods.
4. The performance of ratio of scatter matrices in combination of both LDC and SVC is better in terms of classification accuracy in comparison to other combination of a classifier and feature selection method.
5. The number of features required to obtain maximum classification accuracy is significantly smaller using feature selection methods in comparison to baseline using all classifiers. In particular, the number of features selected in combination of KNN is relatively smaller in comparison to other classifiers. However, the classification accuracy is significantly less in comparison to other classifiers.
6. As the number of features required to obtain maximum classification accuracy is significantly smaller using feature selection methods, the computation time by all the learning methods will be significantly reduced.

Table 1. Variation in Classification Accuracy Different Mental Task

Task	BC	BL	BM	BR	CL	CM	CR	LM	LR	MR	Avg
WFS+ LDC	54.9	53.2	62.2	59.7	57	61.3	57.8	61.1	57	60.6	58.5
Scatt +LDC	92	93.3	97.4	94.6	92	96.3	93	96.1	93.5	94.4	94.3
JC+LDC	90.9	87.5	94.6	91.6	90.9	93.7	88.4	94.6	90.5	89.8	91.3
Reg+LDC	92.8	90.6	96.4	93.2	92.8	96.4	91.3	96	92.8	94.7	93.7
WFS+QDC	49.5	49.9	51.3	49	47.7	48.5	48.5	49.6	49.8	48.3	49.2
Scatt+QDC	91.3	88.3	95	92.9	91.3	94.3	91.5	95.5	91.6	92.5	92.4
JC+QDC	89.9	84.9	91.9	91.1	89.8	93	90.3	93.3	92.6	93.3	91
Reg_+QDC	90.4	85.7	95.4	92.1	90.4	94.1	90.1	95.5	91.7	92.4	91.8
WFS+KNNC	47.7	47	54.3	49.8	54	56	50	54.7	49.4	55.2	51.8
Scatt+KNNC	87	82	88.1	91.1	87	91	86.6	90.6	88	86.6	87.8
JC+KNNC	89.3	82	87.7	92.1	89.3	90.8	86.7	90.6	93.1	89.4	89.1
Reg+KNNC	84.1	79.4	86.4	88.9	84.1	87.8	85	90.5	88.5	86	86.1
WFS+SVC	58.4	59.8	65.1	62	59.7	62.4	63.2	65.8	63.6	66.6	62.7
Scatt+SVC	92.4	93.6	97.3	92.8	92.4	96.3	93.3	96.5	93.3	94.6	94.3
JC+SVC	91.9	88.5	95	93.2	91.9	94.6	91.7	96.5	94.2	94	93.2
Reg+SVC	92.8	90.6	96.6	91.1	92.8	96.4	92.3	96.3	92.8	94.7	93.6

Table 2. Variation of Number of Features required for Different Mental Task

Tasks	BC	B L	B M	B R	C L	C M	C R	L M	L R	M R	Avg
Scatter+LDC	15.3	23.7	21.3	13.8	15.3	19.3	17.7	15.5	14.8	19.3	17.6
JC+LDC	21.3	21.2	20.2	16.3	21.3	17.8	20	12.7	11.7	20	18.3
Reg+LDC	14.2	16.5	14.3	9.7	14.2	16.3	15.3	13.8	12.2	18.3	14.5
Scatter+QDC	12.7	19	17.5	12.2	12.7	19.2	15	14.8	13.3	14.8	15.1
JC+QDC	15.2	14.8	15.8	12.7	15.2	17.8	17	12	9.7	15.8	14.6
Reg+QDC	14.7	15.2	18.2	12.3	14.7	15.5	14.2	11.3	11.3	12.7	14
Scatter+KNNC	3.8	7.7	3.8	6.7	3.8	4.3	2.8	3.3	3.7	5	4.5
JC+KNNC	2.7	3.7	2.2	5.2	2.7	2.7	2.7	3.2	3	4.3	3.2
Reg+KNNC	6.8	7.3	5.5	8	6.8	5.7	5	5	8.2	8	6.6
Scatter+SVC	15.3	24	21.7	11.5	15.3	19.3	14.8	12.3	13.2	19.2	16.7
JC+SVC	21.3	21.2	18.7	12.8	21.3	17.8	22	15.8	14.5	20	18.6
Reg+SVC	13	14.3	13.7	7	13	11.7	13.8	13.7	10.8	13.3	12.4

* WFS=Without feature selection

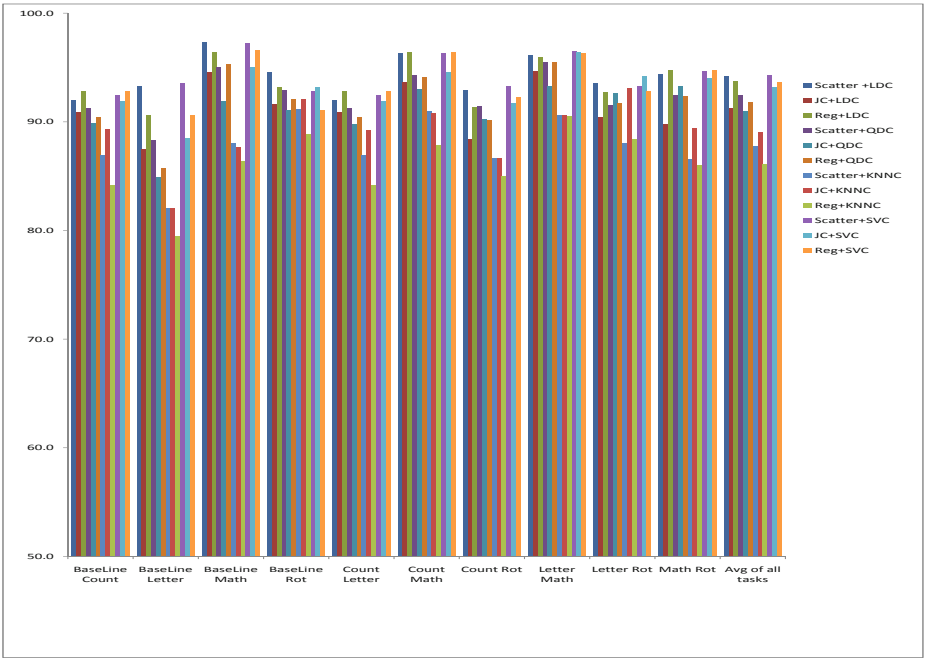


Fig. 1. Variation in Classification Accuracy for Different Mental Task

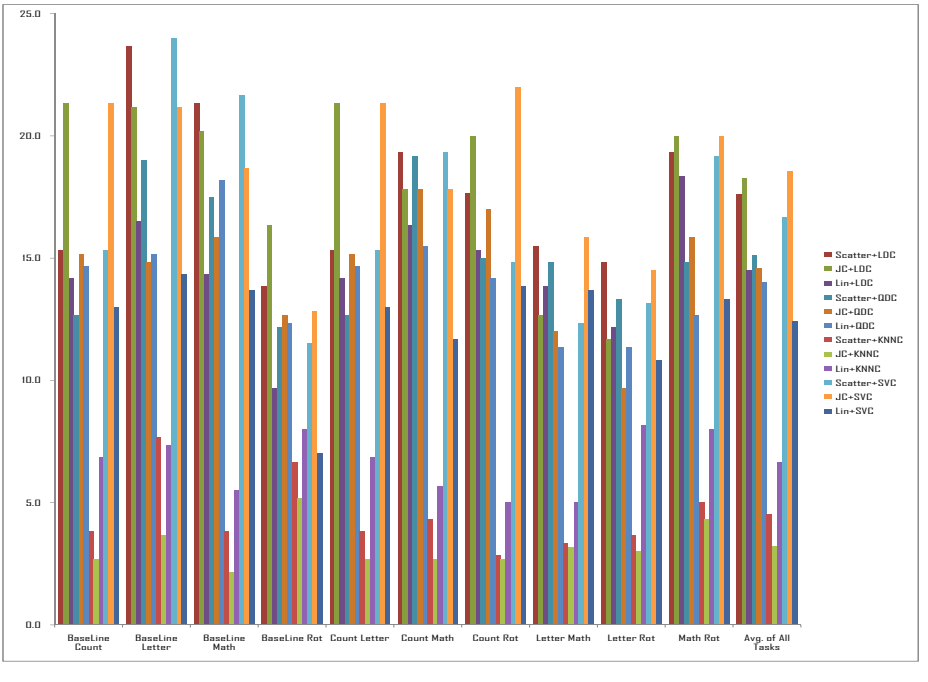


Fig. 2. Variation of Number of Features required for Different Mental Task

5 Conclusion

The performance of a classifier depends on the choice of features and classifier for any pattern recognition system. Features based on Empirical Mode Decomposition from EEG signal is extracted. These features may contain irrelevant and redundant features which makes them inefficient for machine learning. Hence, relevant features which provide maximum classification accuracy are selected using ratio of scatter matrices, Chernoff distance measure and linear regression. The performance of different mental task using different measures used for feature selection is compared and evaluated in terms of classification accuracy. Experimental results show that classification accuracy of all mental tasks classification improve significantly with the use of feature selection methods. In particular the performance of ratio of scatter matrix is better for all classifiers except KNN. The time required to learn the model will decrease significantly as the number of features reduces with the use of feature selections.In future, there is need to develop a feature selection method for mental task classification which gives better performance by all classifiers. It is also required to find out a method of feature extraction which extracts minimal and most relevant features from EEG signal for mental task classification and does not require any further feature selection.

References

1. Anderson, W.C., Stolz, E.A., Shamsunder, S.: Multivariate autoregressive models for classification of spontaneous electroencephalographic signals during mental tasks. IEEE Trans. Biomed. Eng. 45(3), 277–286 (1998)
2. Babiloni, F., Cincotti, F., Lazzarini, L., Millan, J., Mourino, J., Varsta, M., Heikkonen, J., Bianchi, L., Marciani, M.G.: Linear classification of low-resolution EEG patterns produced by imagined hand movements. IEEE Trans. Rehabil. Eng. 8(2), 186–188 (2000)
3. Basseville, M., Benveniste, A.: Sequential segmentation of nonstationary digital signals using spectral analysis. Information Science 29(1), 57–73 (1983)
4. Richard, O., Peter, E., David, G.: Pattern Classification, 2nd edn. Wiley India (P) Ltd
5. Diez, P.F., Mut, V., Laciar, E.: A location of the empirical mode decomposition to the extraction of features from EEG signals for mental task classification. In: 31st Annual Int. Conference of the IEEE EMBS, Minnesota, pp. 2579–2582 (2009)
6. Fisher, A.R.: The use of multiple measurements in taxonomic problems. Ann. Eugen 7, 179–188 (1936)
7. Freeman, W.J.: Comparison of brain models for active vs. passive perception. Information Science 116, 97–107 (1999)
8. Garrett, D., Peterson, D.A., Anderson, C.W., Thaut, M.H.: Comparison of linear, nonlinear, and feature selection methods for EEG signal classification. IEEE Trans. Neural Syst. Rehabil. Eng. 11(2), 141–144 (2003)

9. Graimann, B., Huggins, J.E., Schlogl, A., Levine, S.P., Pfurtscheller, G.: Detection of movement-related desynchronization patterns in ongoing single-channel electrocardiogram. IEEE Trans. Neural Syst. Rehabil. Eng. 11(3), 276–281 (2003)
10. Guyon, I., Elisseeff, A.: An Introduction to Variable and feature Selection. Machine Learning Research (3), 1157–1182 (2003)
11. Guyon, I., Weston, J., Bernhill, S., Vapnik, V.: Gene Selection for cancer classification using support vector machine. Machine Learning (46), 389–422 (2002)
12. Huang, N.E., Shen, Z., Long, S.R., Wu, M.L., Shih, H.H., Zheng, Q., Yen, N.C., Tung, C.C., Liu, H.H.: The empirical mode decomposition and Hilbert spectrum for nonlinear and nonstationary time series analysis. Proc. R. Soc. London A 454, 903–995 (1998)
13. Kauhanen, L., Nykopp, T., Lehtonen, J., Jylanki, P., Heikkonen, J., Rantanen, P., Alaranta, H., Sams, M.: EEG and MEG brain-computer interface for tetraplegic patients. IEEE Trans. Neural Syst. Rehalil. Eng. 14(2), 190–193 (2006)
14. Kohavi, R., John, G.: Wrapper for feature subset selection. Artificial Intelligence (1-2), 273–324 (1997)
15. Lempel, A., Ziv, J.: On the complexity of finite sequences. IEEE Trans. Inform. Theory IT-22, 75–81 (1976)
16. Lingras, P., Butz, C.: Rough set based 1-v-1 and 1-v-r a roaches to support vector machine multi-classification. Information Science 177, 3782–3798 (2007)
17. Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., Arnaldi, B.: A review of classification algorithms for EEG-based brain computer interfaces. Neural Eng. 4, R1–R13 (2007)
18. Muller, K.-R., Anderson, C.W., Birch, G.E.: Linear and non-linear methods for brain computer interfaces. IEEE Trans. Neural Syst. Rehabil. Eng. 11(2), 165–169 (2003)
19. Park, H.-S., Yoo, S.-H.-Y., Cho, S.-B.: Forward selection Method with regression analysis for optimal gene selection in cancer classification. International Journal of Computer Mathematics 84(5), 653–668 (2007)
20. Pfurtscheller, G., Neuper, C., Schlogl, A., Lugger, K.: Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters. IEEE Trans. Rehabil. Eng. 6(3), 316–325 (1998)
21. Pierre, A.D., Kittler, J.: Pattern Recognition: A Statistical Approach. PHI (1982)
22. Ruiz, R., Riquelme, J.C., Ruiz, S.A.: Incremental wrapper based gene selection from microarray data for cancer classification. Pattern Recognition 39(12), 2383–2392 (2006)
23. Shannon, C.E.: A Mathematical Theory of Communication. ATT Tech. J. 27, 379–423, 623–656 (1948)
24. Tibshiran, R., Hastie, T., Narasimha, B., Chu, G.: Diagnosis of multiple cancer types by shrunken centriods of gene expression. Proc. Natl. Acad. Sci., USA (99), 6567–6572 (2002)
25. Tsai, C.-Y.: On detecting nonlinear patterns in discriminate problems. Information Science 176, 772–798 (2006)
26. Tsoi, A.C., So, D.S.C., Sergejew, A.: Classification of electroencephalogram using artificial neural networks. In: Advances Neural Information Processing Systems, vol. 6, pp. 1151–1158. Morgan Kaufman, San Francisco (1994)
27. Yom-Tov, E., Inbar, G.F.: Feature selection for the classification of movements from single movement-related potentials. IEEE Trans. Neural Syst. Rehabil. Eng. 10(3), 170–177 (2002)

28. Zachary, A.K., Jorge, I.A.: A new mode of communication between man and his surroundings. IEEE Transactions on Biomedical Engineering 37(12), 1209–1214 (1990)
29. Zachary, A.K.: Alternative modes of communication between man and machine. Master's thesis, Purdue University (1988)
30. Zhang, X.S., Roy, R.J., Jensen, E.W.: EEG complexity as a measure of depth anesthesia for patients. IEEE Trans. Biomed. Eng. (48), 1424–1433 (2001)
31. Zhou, S.M., Gan, J.Q.: Constructing parsimonious fuzzy classifiers based on L2-SVM in high-dimensional space with automatic model selection and fuzzy rule ranking. IEEE Trans. Fuzzy Syst. 15(3), 398–409 (2007)

Author Index