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

LNAI 7301

Advances in Knowledge Discovery and Data Mining

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

1 Part I

Springer

Lecture Notes in Artificial Intelligence 7301

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 I

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 978-3-642-30216-9 e-ISBN 978-3-642-30217-6
DOI 10.1007/978-3-642-30217-6
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)
Wen Cheong Chin (Co-chair)
Nadim Jahangir
Chee Onn Wong
Chong Pei Fen
Timothy Yap
Kok Leong Chan
Azurawati
Khong Leng Lim
Michelle
Kean Vee Sor
Madhavan
Chin Chwee Wong

Victor Tan (Co-chair)
Kok Why Ng (Co-chair)
Choo Yee Ting
Chiung Ching Ho
Hau Lee Tong
James Ooi
Yong Haur Tay
Chian Wen Too
Mariam
Meei Hao Hoo
Priya
Simon Lau
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
AF 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 I

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

Unsupervised Learning: Clustering, Probabilistic Modeling

Table of Contents – Part II

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

Data Manipulation: Pre-processing and Dimension Reduction

Time-Evolving Relational Classification and Ensemble Methods

Ryan Rossi and Jennifer Neville

Purdue University,
West Lafayette, IN 47906, USA
{rrossi,neville}@purdue.edu

Abstract. Relational networks often evolve over time by the addition, deletion, and changing of links, nodes, and attributes. However, accurately incorporating the full range of temporal dependencies into relational learning algorithms remains a challenge. We propose a novel framework for discovering *temporal-relational representations* for classification. The framework considers transformations over *all* the evolving relational components (attributes, edges, and nodes) in order to accurately incorporate temporal dependencies into relational models. Additionally, we propose *temporal ensemble methods* and demonstrate their effectiveness against traditional and relational ensembles on two real-world datasets. In all cases, the proposed temporal-relational models outperform competing models that ignore temporal information.

1 Introduction

Temporal-relational information is present in many domains such as the Internet, citation and collaboration networks, communication and email networks, social networks, biological networks, among many others. These domains all have attributes, links, and/or nodes changing over time which are important to model. We conjecture that discovering an accurate *temporal-relational representation* will disambiguate the true nature and strength of links, attributes, and nodes. However, the majority of research in relational learning has focused on modeling static snapshots [2,6] and has largely ignored the utility of learning and incorporating temporal dynamics into relational representations.

Temporal relational data has three main components (attributes, nodes, links) that vary in time. First, the attribute values (on nodes or links) may change over time (e.g., research area of an author). Next, links might be created and deleted throughout time (e.g., host connections are opened and closed). Finally, nodes might appear and disappear over time (e.g., through activity in an online social network).

Within the context of evolving relational data, there are two types of prediction tasks. In a *temporal* prediction task, the attribute to predict is changing over time (e.g., student GPA), whereas in a *static* prediction task, the predictive attribute is constant (e.g., paper topic). For these prediction tasks, the space of temporal-relational representations is defined by the set of relational

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 1–13, 2012.

elements that change over time (attributes, links, and nodes). To incorporate temporal information in a representation that is appropriate for relational models, we consider two transformations based on *temporal weighting* and *temporal granularity.* Temporal weighting aims to represent the temporal influence of the links, attributes and nodes by decaying the weights of each with respect to time, whereas the choice of temporal granularity restricts attention to links, attributes, and nodes within a particular window of time. The optimal temporal-relational representation and the corresponding temporal classifier depends on the particular temporal dynamics of the links, attributes, and nodes present in the data, as well as the network domain (e.g., social vs. biological networks).

In this work, we address the problem of selecting the most optimal temporal-relational representation to increase the accuracy of predictive models. We consider the full space of *temporal-relational representations* and propose **(1)** a temporal-relational classification framework, and **(2)** a set of temporal ensemble methods, to leverage time-varying links, attributes, and nodes in relational networks. We illustrate the different types of models on a variety of classification tasks and evaluate each under various conditions. The results demonstrate the flexibility and effectiveness of the temporal-relational framework for classification in time-evolving relational domains. Furthermore, the framework provides a foundation for automatically searching over temporal-relational representations to increase the accuracy of predictive models.

2 Related Work

Recent work has started to model network dynamics in order to better predict link and structure formation over time [7, 10], but this work focuses on unattributed graphs. Previous work in relational learning on attributed graphs either uses static network snapshots or significantly limits the amount of temporal information incorporated into the models. Sharan et al. [18] assumes a strict representation that only uses kernel estimation for link weights, while GA-TVRC [9] uses a genetic algorithm to learn the link weights. SRPTs [11] incorporate temporal and spatial information in the relational attributes. However, the above approaches focus only on one specific temporal pattern and do not consider different temporal granularities. In contrast, we explore a larger space of temporal-relational representations in a flexible framework that can capture temporal dependencies over *links*, *attributes*, and *nodes*.

To the best of our knowledge, we are the first to propose and investigate *temporal-relational ensemble methods* for time-varying relational classification. However, there has been recent work on relational ensemble methods [8, 14, 15] and non-relational ensemble methods for evolving streams [1]. Preisach et al. [14] use voting and stacking methods to combine relational data with multiple relations. In contrast, Eldardiry and Neville [8] incorporates prediction averaging in the collective inference process to reduce both learning and inference variance.

3 Temporal-Relational Classification Framework

Below we outline a temporal-relational classification framework for prediction tasks in dynamic relational networks. Relational data is represented as an attributed graph $D = (G, \mathbf{X})$ where the graph $G = (V, E)$ represents a set of N nodes, such that $v_i \in V$ corresponds to node i and each edge $e_{ij} \in E$ corresponds to a link (e.g., email) between nodes i and j. The attribute set:

$$\mathbf{X} = \begin{pmatrix} \mathbf{X^V} = [X^1, X^2, ..., X^{m_v}], \\ \mathbf{X^E} = [X^{m_v+1}, X^{m_v+2}, ..., X^{m_v+m_e}] \end{pmatrix}$$

contains m_v observed attributes on the nodes ($\mathbf{X^V}$) and m_e observed attributes on the edges ($\mathbf{X^E}$). Dynamic relational data evolves over time by the addition, deletion, and changing of nodes, edges, and attributes. Let $D_t = (G_t, \mathbf{X}_t)$ refer to the dataset at time t, where $G_t = (V, E_t)$ and $\mathbf{X}_t = (\mathbf{X}_t^V, \mathbf{X}_t^E)$. In our classification framework, we consider relational data observed over a range of timesteps $t = \{1, ..., T\}$ (e.g., citations over a period of years, emails over a period of days). Given this time-varying relational data, the task is to learn a model to predict either a static attribute Y or a dynamic attribute at a particular timestep Y_t, while exploiting both the relational and temporal dependencies in the data.

We define our temporal-relational classification framework with respect to a set of possible transformations of links, attributes, or nodes (as a function of time). The temporal weighting (e.g., exponential decay of past information) and temporal granularity (e.g., window of timesteps) of the links, attributes and nodes form the basis for any arbitrary transformation with respect to the temporal information (See Table 1). The discovered temporal-relational representation can be applied for mining temporal patterns, classification, and as a means for constructing temporal-ensembles. An overview of the temporal-relational representation discovery is provided below:

Table 1. Temporal-Relational Representation

1. For each Relational Component
   - Links, Attributes, or Nodes
2. Select the Temporal Granularity
   - ⋆ Timestep t_i
   - ⋆ Window $\{\mathbf{t_j}, \mathbf{t_{j+1}}, ..., \mathbf{t_i}\}$
   - ⋆ Union $T = \{t_0, ..., t_n\}$
3. Select the Temporal Influence
   - ⋆ Weighted
   - ⋆ Uniform

   Repeat steps 1-3 for each component.
4. Select the Relational Classifier
   - ⋆ Relational Bayes Classifier (RBC)
   - ⋆ Relational Probability Trees (RPT)

	Uniform			**Weighting**		
	Timestep	**Window**	**Union**	**Timestep**	**Window**	**Union**
Edges						
Attributes						
Nodes						

Table 1 provides an intuitive view of the possible temporal-relational representations. For instance, the TVRC model is a special case of the proposed framework where the links, attributes, and nodes are unioned and the links are weighted. Below we provide more detail on steps 2-4.

3.1 Temporal Granularity

Traditionally, relational classifiers have attempted to use all the data available in a network [18]. However, since the relevance of data may change over time (e.g., links become stale), learning the *appropriate* temporal granularity (i.e., range of timesteps) can improve classification accuracy. We briefly define three general classes for varying the temporal granularity of the links, attributes, and nodes.

1. **Timestep.** The timestep models only use a single timestep t_i for learning.
2. **Window.** The window models use a sliding window of (multiple) timesteps $\{t_j, t_{j+1}, ..., t_i\}$ for learning. When the size of window is varied, the space of possible models in this category is by far the largest.
3. **Union.** The union model uses all previous temporal information for learning at time t_i, i.e., $T = \{0, ..., t_i\}$.

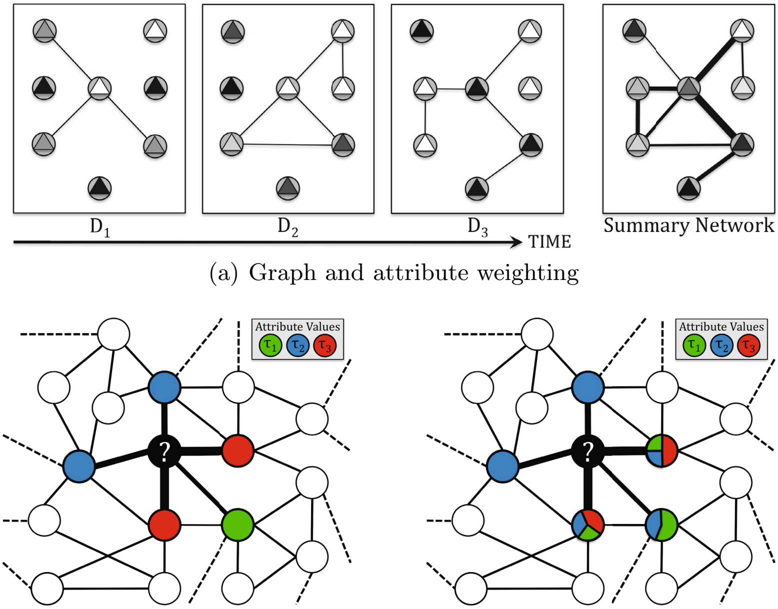
The timestep and union models are separated into distinct classes for clarity in evaluation and for understandability in pattern mining.

3.2 Temporal Influence: Links, Attributes, Nodes

We model the influence of relational components over time using temporal weighting. Specifically, when considering a temporal dataset $D_t = (G_t, \mathbf{X}_t)$, we will construct a weighted network $G_t = (V, E_t, W_t^E)$ and $\mathbf{X}_t = (\mathbf{X}_t^V, \mathbf{X}_t^E, W_t^X)$. Here W_t refers to a function that assigns weights on the edges and attributes that are used in the classifiers below.

Initially, we define $W_t^E(i,j) = 1$ if $e_{ij} \in E_t$ and 0 otherwise. Similarly, we define $W_t^X(x_i^m) = 1$ if $X_i^m = x_i^m \in \mathbf{X_t^m}$ and 0 otherwise. Then we consider two different approaches to revise these initial weights:

1. **Weighting.** These temporal weights can be viewed as probabilities that a relational component is still active at the current time step t, given that it was observed at time $(t-k)$. We investigated three temporal weighting functions:
   - *Exponential Kernel.* The exponential kernel weights the recent past highly and decays the weight rapidly as time passes [3]. The kernel function K_E for temporal data is defined as: $K_E(D_i; t, \theta) = (1-\theta)^{t-i}\theta W_i$
   - *Linear Kernel.* The linear kernel decays more grdually and retains the historical information longer: $K_L(D_i; t, \theta) = \theta W_i(\frac{t_* - t_i + 1}{t_* - t_o + 1})$
   - *Inverse Linear Kernel.* This kernel lies between the exponential and linear kernels when moderating historical information: $K_{IL}(D_i; t, \theta) = \theta W_i(\frac{1}{t_i - t_o + 1})$



(a) Graph and attribute weighting

(b) Incorporating link weights

(c) Using link & attribute weights

Fig. 1. (a) Temporally weighting the attributes and links. **(b)** The feature calculation that includes only the temporal link weights. **(c)** The feature calculation that incorporates *both* the temporal attribute weights and the temporal link weights.

2. **Uniform.** These weights ignore the temporal influence of a relational component, and weight them uniformly over time, i.e., $W_t^E(i,j) = 1$ if $e_{ij} \in E_{t'} : t' \in T$ and 0 otherwise. A relational component can be assigned uniform weights within the selected temporal granularity or over the entire time window (e.g., traditional classifiers assign uniform weights, but they don't select the appropriate temporal granularity).

We note that different weighting functions can be chosen for different relational components (edges, attributes, nodes) with varying temporal granularities. For instance, the temporal influence of the links might be predicted using the exponential kernel while the attributes are uniformly weighted but have a different temporal granularity than the links.

3.3 Temporal-Relational Classifiers

Once the temporal granularity and temporal weighting are selected for each relational component, then a temporal-relational classifier can learned. In this work, we use modified versions of the RBC [13] and RPT [12] to model the transformed temporal-relational representation. However, we note that any relational model that can be modified to incorporate node, link, and attribute weights is suitable for this phase. We extended RBCs and RPTs since they are interpretable, diverse, simple, and efficient. We use k-fold x-validation to learn the "best" model. Both classifiers are extended for learning and prediction over time.

Weighted Relational Bayes Classifier. RBCs extend naive Bayes classifiers [5] to relational settings by treating heterogeneous relational subgraphs as a homogeneous set of attribute multisets. The weighted RBC uses standard maximum likelihood learning. More specifically, the sufficient statistics for each conditional probability distribution are computed as weighted sums of counts based on the link and attribute weights. More formally, for a class label C, attributes $\mathbf{X}$, and related items R, the RBC calculates the probability of C for an item i of type $G(i)$ as follows:

$$P(C^i|\mathbf{X}, R) \propto \prod_{X_m \in \mathbf{X}^{\mathbf{G(i)}}} P(X_m^i|C) \prod_{j \in R} \prod_{X_k \in \mathbf{X}^{\mathbf{G(j)}}} P(X_k^j|C)P(C)$$

Weighted Relational Probability Trees. RPTs extend standard probability estimation trees to a relational setting. We use the standard learning algorithm [12] except that the aggregate functions are computed after the appropriate links and attributes weights are included for the selected temporal granularity (shown in Figure 1). For prediction, if the model is applied to predict attribute Y_t at time t, we first calculate the weighted data D_t . Then the learned model from time $(t-1)$ is applied to D_t. The weighted classifier is appropriately augmented to incorporate the weights from D_t.

4 Temporal Ensemble Methods

Ensemble methods have traditionally been used to improve predictions by considering a weighted vote from a set of classifiers [4]. We propose temporal ensemble methods that exploit the *temporal dimension of relational data* to construct more accurate predictors. This is in contrast to traditional ensembles that do not explicitly use the temporal information. The *temporal-relational classification framework* and in particular the temporal-relational representations of the time-varying links, nodes, and attributes form the basis of the temporal ensembles (i.e., as a wrapper over the framework). The proposed temporal ensemble techniques are drawn from one of the five methodologies described below.

1. **Transforming the Temporal Nodes and Links:** The first method learns an ensemble of classifiers, where each of the classifiers are learned from, and then applied to, link and node sets that are sampled from each discrete timestep according to some probability. This sampling strategy is performed after selecting a temporal weighting and temporal granularity, and transforming the data to the appropriate temporal-relational representation. We note that the sampling probabilities for each timestep can be modified to bias the sampling toward the present or the past.

2. **Sampling or Transforming the Temporal Feature Space:** The second method transforms the temporal feature space by localizing randomization (for attributes at each timestep), weighting, or by varying the temporal granularity of the features, and then learning an ensemble of classifiers with different feature sets. Additionally, we might use only one temporal weighting function but learn models with different decay parameters or resample from the temporal features.
3. **Adding Noise or Randomness:** The third method is based on adding noise along the temporal dimension of the data, to increase generalization and performance. Specifically, we randomly permute the nodes feature values across the timesteps (i.e., a nodes recent behavior is observed in the past and vice versa) or links between nodes are permuted across time, and then learn an ensemble of models from several versions of the data.
4. **Transforming the Time-Varying Class Labels:** The fourth method introduces variance in the data by randomly permuting the previously learned labels at t-1 (or more distant) with the true labels at t, again learning an ensemble of models from several versions of the data.
5. **Multiple Classification Algorithms and Weightings:** The fifth method constructs and ensemble by randomly selecting from a set of classification algorithms (i.e., RPT, RBC, wvRN, RDN), while using the same temporal-relational representation, or by varying the representation with respect to the temporal weighting or granularity. Notably, an ensemble that uses both RPT and RBC models significantly increases accuracy, most likely due to the diversity of these temporal classifiers (i.e., correctly predicting different instances). Additionally, the temporal-classifiers might be assigned weights based on assessment of accuracy from cross-validation (or a Bayesian model selection approach).

5 Methodology

For evaluating the framework, we use both static (Y is constant over time) and temporal prediction tasks (Y_t changes over time).

5.1 Datasets

PyComm Developer Communication Network. We analyze email and bug communication networks extracted from the python-dev mailing list archive (www.python.org) for the period 01/01/07–09/30/08. The network consists of 13181 email messages, among 1914 users. Bug reports were also extracted and used to construct a *bug* discussion network consisting of 69435 bug comments among 5108 users. The size of the timesteps are three months. We also extracted text from emails and bug messages and use it to dynamically model topics between individuals and teams. Additionally, we discover temporal centrality attributes (i.e., clustering coefficient, betweenness). The prediction task is whether a developer is effective (i.e., if a user closed a bug in that timestep).

Cora Citation Network. The CORA dataset contains authorship and citation information about CS research papers extracted automatically from the web. The prediction tasks are to predict one of seven machine learning papers and to predict AI papers given the topic of its references. In addition, these techniques are evaluated using the most prevalent topics its authors are working on through collaborations with other authors.

5.2 Temporal Models

The space of temporal-relational models are evaluated using a representative sample of classifiers with varying temporal weightings and granularities. For every timestep t, we learn a model on D_t (i.e., some set of timesteps) and apply the model to D_{t+1}. The utility of the temporal-relational classifiers and representation are measured using the area under the ROC curve (AUC). Below, we briefly describe a few classes of models that were evaluated.

- **TENC:** The TENC models predict the temporal influence of both the links and attributes [16].
- **TVRC:** This model weights only the links using all previous timesteps.
- **Union Model:** The union model uses all links and nodes up to and including t for learning.
- **Window Model:** The window model uses the data D_{t-1} for prediction on D_t (unless otherwise specified).

We also compare simpler models such as the RPT (relational information only) and the DT (non-relational) that ignore any temporal information. Additionally, we explore many other models, including the class of window models, various weighting functions (besides exponential kernel), and built models that vary the set of windows in TENC and TVRC.

6 Empirical Results

In this section, we demonstrate the effectiveness of the temporal-relational framework and temporal ensemble methods on two real-world datasets. The main findings are summarized below:

- ⋆ Temporal-relational models significantly outperform relational and non-relational models.
- ⋆ The classes of temporal-relational models each have advantages and disadvantages in terms of accuracy, efficiency, and interpretability. Models based strictly on temporal granularity are more interpretable but less accurate than models that *learn* the temporal influence. The more complex models that combine both are generally more accurate, but less efficient.
- ⋆ *Temporal ensemble methods* significantly outperform non-relational and relational ensembles. In addition, the temporal ensembles are an efficient and accurate alternative to searching over the space of temporal models.

6.1 Single Models[1]

We evaluate the temporal-relational framework using single-models and show that in all cases the performance of classification improves when the temporal dynamics are appropriately modeled.

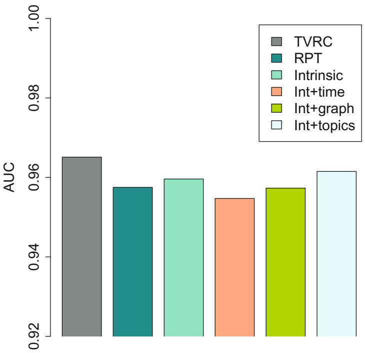


Fig. 2. Comparing a primitive *temporal* model (TVRC) to competing relational (RPT), and non-relational (DT) models

Temporal, Relational, and Non-Relational Information. The utility of the temporal (TVRC), relational (RPT), and non-relational information (decision tree; DT) is assessed using the most primitive models. Figure 2 compares TVRC with the RPT and DT models that use more features but ignore the temporal dynamics of the data. We find the TVRC to be the simplest temporal-relational classifier that still outperforms the others. Interestingly, the discovered topic features are the only additional features that improve performance of the DT model. This is significant as these attributes are discovered by dynamically modeling the topics, but are included in the DT model as simple non-relational features (i.e., no temporal weighting or granularity).

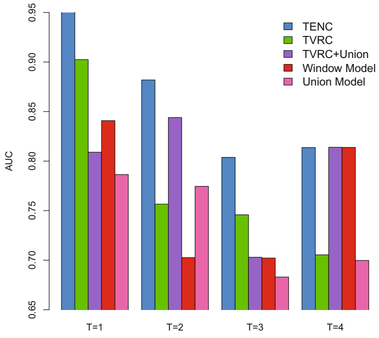


Fig. 3. Exploring the space of temporal relational models. Significantly different temporal-relational representations from the proposed framework are evaluated.

Exploring Temporal-Relational Models. We focus on exploring a representative set of temporal-relational models from the proposed framework. To more appropriately evaluate the models, we remove highly correlated attributes (i.e., that are not necessarily temporal patterns, or motifs), such as "assignedto" in the PyComm prediction task. In Figure 3, we find that TENC outperforms the other models over all timesteps. This class of models are significantly more complex than TVRC since the temporal influence of both links and attributes are learned.

We then explored learning the appropriate temporal granularity. Figure 3 shows the results from two models in the TVRC class where we tease apart the superiority of TENC (i.e., weighting or granularity). However, both TVRC models outperform one another on different timesteps, indicating the necessity for a more precise temporal-representation that optimizes the temporal granularity by selecting the appropriate decay parameters for links and attributes

[1] For brevity, some plots and comparisons were omitted [17].

(i.e., TENC). Similar results were found using Cora and other base classifiers such as RBC. Models based strictly on varying the *temporal granularity* were also explored. More details can be found in [17].

6.2 Temporal-Ensemble Models

Instead of directly learning the optimal temporal-relational representation to increase the accuracy of classification, we use *temporal ensembles* by varying the relational representation with respect to the temporal information. These ensemble models reduce error due to variance and allow us to assess which features are most relevant to the domain with respect to the relational or temporal information.

Temporal, Relational, and Traditional Ensembles. We first resampled the instances (nodes, links, features) repeatedly and then learn TVRC, RPT, and DT models. Across almost all the timesteps, we find the temporal-ensemble that uses various temporal-relational representations outperforms the relational-ensemble and the traditional ensemble (see Figure 4). The temporal-ensemble outperforms the others even when the minimum amount of temporal information is used (e.g., time-varying links). More sophisticated temporal-ensembles can be constructed to further increase accuracy. We have investigated ensembles that use significantly different temporal-relational representations (i.e., from a wide range of model classes) and ensembles that use various temporal weighting parameters. In all cases, these ensembles are more robust and increase the accuracy over more traditional ensemble techniques (and single classifiers). Further, the average improvement of the temporal-ensembles is significant at $p < 0.05$ with a 16% reduction in error, justifying the proposed temporal ensemble methodologies.

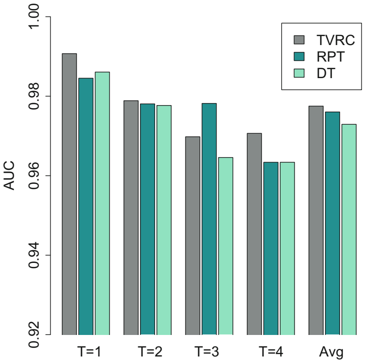


Fig. 4. Comparing temporal, relational, and traditional ensembles

In the next experiment, we construct ensembles using the feature classes. We use the primitive models (with the transformed feature space) in order to investigate (more accurately) the most significant feature class (communication, team, centrality, topics) and also to identify the minimum amount of temporal information required to outperform relational ensembles.

In Figure 5, we find several striking temporal patterns. First, the team features are localized in time and are not changing frequently. For instance, it is unlikely that a developer changes their assigned teams and

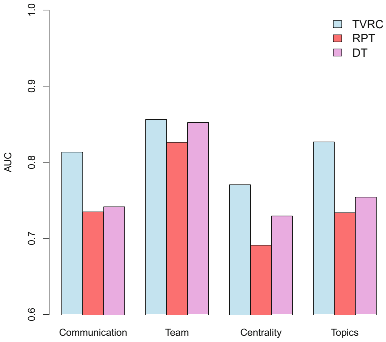


Fig. 5. Comparing attribute classes w.r.t. temporal, relational, and traditional ensembles

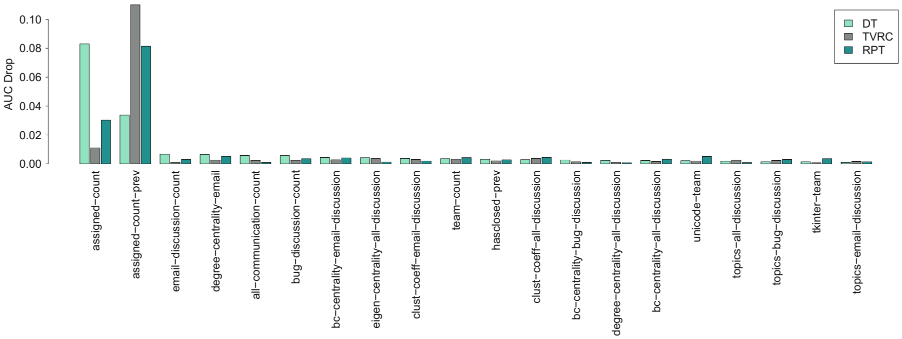


Fig. 6. Randomization. The significant attributes used in the *temporal ensemble* are compared to the relational and traditional ensembles. The change in AUC is measured.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
dev	logged	gt	code	test
wrote	patch	file	object	lib
guido	issue	lt	class	view
import	bugs	line	case	svn
code	bug	os	method	trunk
pep	problem	import	type	rev
mail	fix	print	list	modules
release	fixed	call	set	build
tests	days	read	objects	amp
work	created	socket	change	error
people	time	path	imple	usr
make	docu	data	functions	include
pm	module	error	argument	home
ve	docs	open	dict	file
support	added	windows	add	run
module	check	problem	def	main
things	doc	traceback	methods	local
good	doesnt	mailto	exception	src
van	report	recent	ms	directory

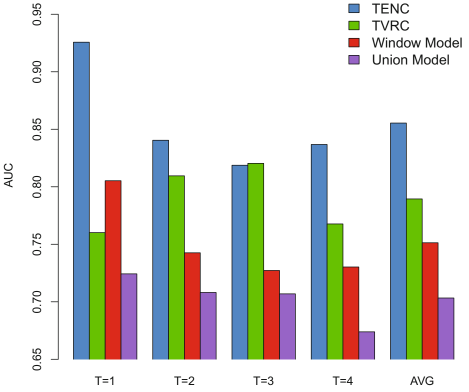


Fig. 7. Evaluation of temporal-relational classifiers using only the latent topics of the communications to predict effectiveness. LDA is used to automatically discover the latent topics as well as annotating the communication links and individuals with their appropriate topic in the temporal networks.

therefore modeling the temporal dynamics only increases accuracy by a relatively small percent. However, the *temporal-ensemble* is still more accurate than traditional ensemble methods that ignore temporal patterns. This indicates the robustness of the temporal-relational representations. More importantly, the other classes of attributes are evolving considerably and this fact is captured by the significant improvement of the temporal ensemble models. Similar performance is also obtained by varying the temporal granularity (see previous examples).

Randomization. We use randomization to identify the significant attributes in the *temporal-ensemble models.* Randomization provides a means to rank and eliminate redundant attributes (i.e., two attributes may share the same significant temporal pattern). We randomize each attribute in each timestep and measure the change in AUC. The results are shown in Figure 6.

We find that the basic traditional ensemble relies on “assignedto” (in the current time step) while the temporal ensemble (and even less for the relational ensemble) relies on the previous “assignedto” attributes. This indicates that relational information in the past is more useful than intrinsic information in the present—which points to an interesting hypothesis that a colleagues behavior (and interactions) precedes their own behavior. Organizations might use this to predict future behavior with less information and proactively respond more quickly. Additionally, the topic attributes are shown to be the most useful for the temporal ensembles (Fig. 7), indicating the utility of using topics to understand the context and strength of relationships.

7 Conclusion

We proposed and validated a framework for temporal-relational classifiers, ensembles, and more generally, representations for temporal-relational data. We evaluated an illustrative set of temporal-relational models from the proposed framework. Empirical results show that the models significantly outperform competing classification models that use either no temporal information or a very limited amount. The proposed temporal ensemble methods (i.e., temporally sampling, randomizing, and transforming features) were shown to significantly outperform traditional and relational ensembles. Furthermore, the temporal-ensemble methods were shown to increase the accuracy over traditional models while providing an efficient alternative to exploring the space of temporal-models. The results demonstrated the effectiveness, scalability, and flexibility of the temporal-relational representations for classification and ensembles in time-evolving domains. In future work, we will theoretically analyze the framework and the proposed ensemble methods.

Acknowledgments. This research is supported by NSF under contract number SES-0823313 and by the Department of Defense through a National Defense Science & Engineering Graduate Fellowship (NDSEG). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of the NSF or the U.S. Government.

References

1. Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., Gavaldà, R.: New ensemble methods for evolving data streams. In: SIGKDD, pp. 139–148 (2009)
2. Chakrabarti, S., Dom, B., Indyk, P.: Enhanced hypertext categorization using hyperlinks. In: SIGMOD, pp. 307–318 (1998)
3. Cortes, C., Pregibon, D., Volinsky, C.: Communities of Interest. In: Hoffmann, F., Adams, N., Fisher, D., Guimarães, G., Hand, D.J. (eds.) IDA 2001. LNCS, vol. 2189, pp. 105–114. Springer, Heidelberg (2001)

4. Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
5. Domingos, P., Pazzani, M.: On the optimality of the simple bayesian classifier under zero-one loss. Machine Learning 29, 103–130 (1997)
6. Domingos, P., Richardson, M.: Mining the network value of customers. In: SIGKDD, pp. 57–66 (2001)
7. Dunlavy, D., Kolda, T., Acar, E.: Temporal link prediction using matrix and tensor factorizations. TKDD 5(2), 10 (2011)
8. Eldardiry, H., Neville, J.: Across-model collective ensemble classification. AAAI (2011)
9. Güneş, İ., Çataltepe, Z., Öğüdücü, Ş.G.: GA-TVRC: A Novel Relational Time Varying Classifier to Extract Temporal Information Using Genetic Algorithms. In: Perner, P. (ed.) MLDM 2011. LNCS, vol. 6871, pp. 568–583. Springer, Heidelberg (2011)
10. Lahiri, M., Berger-Wolf, T.: Structure prediction in temporal networks using frequent subgraphs. In: CIDM, pp. 35–42 (2007)
11. McGovern, A., Collier, N., Matthew Gagne, I., Brown, D., Rodger, A.: Spatiotemporal Relational Probability Trees: An Introduction. In: ICDM, pp. 935–940 (2008)
12. Neville, J., Jensen, D., Friedland, L., Hay, M.: Learning relational probability trees. In: SIGKDD, pp. 625–630 (2003)
13. Neville, J., Jensen, D., Gallagher, B.: Simple estimators for relational Bayesian classifers. In: ICML, pp. 609–612 (2003)
14. Preisach, C., Schmidt-Thieme, L.: Relational ensemble classification. In: ICDM, pp. 499–509. IEEE (2006)
15. Preisach, C., Schmidt-Thieme, L.: Ensembles of relational classifiers. KIS 14(3), 249–272 (2008)
16. Rossi, R., Neville, J.: Modeling the evolution of discussion topics and communication to improve relational classification. In: SOMA-KDD, pp. 89–97 (2010)
17. Rossi, R.A., Neville, J.: Representations and ensemble methods for dynamic relational classification. CoRR abs/1111.5312 (2011)
18. Sharan, U., Neville, J.: Temporal-relational classifiers for prediction in evolving domains. In: ICML (2008)

Active Learning for Hierarchical Text Classification

Xiao Li, Da Kuang, and Charles X. Ling

Department of Computer Science,
The University of Western Ontario, London, Ontario, N6A 5B7, Canada
{xli485,dkuang,cling}@csd.uwo.ca

Abstract. Hierarchical text classification plays an important role in many real-world applications, such as webpage topic classification, product categorization and user feedback classification. Usually a large number of training examples are needed to build an accurate hierarchical classification system. Active learning has been shown to reduce the training examples significantly, but it has not been applied to hierarchical text classification due to several technical challenges. In this paper, we study active learning for hierarchical text classification. We propose a realistic multi-oracle setting as well as a novel active learning framework, and devise several novel leveraging strategies under this new framework. Hierarchical relation between different categories has been explored and leveraged to improve active learning further. Experiments show that our methods are quite effective in reducing the number of oracle queries (by 74% to 90%) in building accurate hierarchical classification systems. As far as we know, this is the first work that studies active learning in hierarchical text classification with promising results.

1 Introduction

Hierarchical text classification plays an important role in many real-world applications, such as webpage topic classification, product categorization and user feedback classification. Due to the rapid increase of published documents (e.g., articles, patents and product descriptions) online, most of the websites (from Wikipedia and Yahoo! to the small enterprise websites) classify their documents into a predefined hierarchy (or taxonomy) for easy browsing. As more documents are published, more human efforts are needed to give the hierarchical labels of the new documents. It dramatically increases the maintenance cost for those organization or companies. To tackle this problem, machine learning techniques such as hierarchical text classification can be utilized to automatically categorize new documents into the predefined hierarchy.

Many approaches have been proposed to improve the performance of hierarchical text classification. Different approaches have been proposed in terms of how to build the classifiers [2,19], how to construct the training sets [1,6] and how to choose the decision thresholds [15,1] and so on. As a hierarchy may contain hundreds or even tens of thousands of categories, those approaches often

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 14–25, 2012.

require a large number of labeled examples for training. However, in real-world applications, such as webpage topic classification, the *labeled* documents are very limited compared to the total number of *unlabeled* documents. Obtaining a large size of labeled documents for training requires great amount of human efforts. How can we build a reliable hierarchical classifier from a relatively small number of examples? Can we reduce the number of labeled examples significantly?

To tackle the lack of labeled examples, active learning can be a good choice [16,12,18]. The idea of active learning is that, instead of passively receiving the training examples, the learner actively selects the most "informative" examples for the current classifier and gets their labels from the oracle (i.e., human expert). Usually, those most informative examples can benefit the classification performance most. Several works have successfully applied active learning in text classification [16,5,20]. However, to our best knowledge, no previous works have been done in hierarchical text classification with active learning due to several technical challenges. For example, as a large taxonomy can contain thousands of categories, it is impossible to have one oracle to provide all labels. Thus, similar to DMOZ[1], multiple oracles are needed. What would be a realistic setting for multiple oracles for active learning in hierarchical text classification? How can we leverage the hierarchical relation to further improve active learning?

In this paper, we study how active learning can be effectively applied to hierarchical text classification so that the number of labeled examples (or oracle queries) needed can be reduced significantly. We propose a new setting of multiple oracles, which is currently in use in many real-world applications (e.g., DMOZ). Based on this setting, we propose an effective framework for active learning in hierarchical text classification. Moreover, we explore how to utilize the hierarchical relation to further improve active learning. Accordingly, several leveraging strategies and heuristics are devised. According to our experiments, active learning under our framework significantly outperforms the baseline learner, and the additional strategies further enhance the performance of active learning for hierarchical text classification. Compared to the best performance of the baseline hierarchical learner, our best strategy can reduce the number of oracle queries by 74% to 90%.

2 A Novel Multi-oracle Setting

When active learning is applied to text classification, as far as we know, all previous works (e.g., [5,20]) explicitly or implicitly assume that given a document that might be associated with multiple labels, there always exist oracles who can perfectly answer all labels. In hierarchical text classification, it is very common that the target hierarchy has a large number of categories (e.g., DMOZ has over one million categories) across various domains, and thus it is unrealistic for one oracle (expert) to be "omniscient" in everything. For example, an expert in "Business" may have less confidence about "Computer", and even less about

[1] It is often called Open Directory Project (`http://www.dmoz.org`).

"Programming". If the expert in "Business" has to label "Programming", errors can occur. Such error introduces noise to the learner.

Therefore, it is more reasonable to assume that there are multiple oracles who are experts in different domains. Each oracle only gives the label(s) related to his or her own domains. Thus, the labels provided by multiple oracles will be more accurate and reliable than the labels given by only one oracle. Although previous works have studied active learning with multiple oracles [4,10], as far as we know, their settings are quite different from ours as their oracles provide labels for all examples for only one category, while in our case, different oracles provide labels for examples in different categories in the hierarchy.

Our setting of multiple oracles is actually implemented in DMOZ. As far as we know, DMOZ holds a large number of categories. Each category is generally maintained by at least one human editor whose responsibility is to decide whether or not a submitted website belongs to that category.[2] We adopt the similar setting of DMOZ. In our setting, each category in the hierarchy has one oracle, who decides solely if the selected document belongs to the current category or not (by answer "Yes" or "No").

3 A New Framework of Hierarchical Active Learning

In this paper, we mainly discuss pool-based active learning where a large pool of unlabeled examples is available for querying oracles. Figure 1 shows the basic idea of our hierarchical active learning framework. Simply speaking, at each iteration of active learning, classifiers on different categories *independently* and *simultaneously* select the most informative examples from the unlabeled pool for themselves, and ask the oracles on the corresponding categories for the labels. The major steps of our hierarchical active learning algorithm are as follows:

1. We first train a binary classifier (C) on each category to distinguish it from its sibling categories. The training set (D^L) is constructed by using the positive examples from the training set of the parent category [14].[3]
2. Then, we construct the local unlabeled pool (D^U) for each classifier (see Section 3.1), select the most informative examples from the local unlabeled pool for that classifier, and query the corresponding oracle for the labels.
3. For each query, the oracle returns "Yes" or "No" to indicate whether the queried example belongs to that category or not. Based on the answers, the classifier updates its classification model (see Section 3.2).
4. This process is executed simultaneously on all categories at each iteration and repeats until the terminal condition is satisfied.

There are two key steps (step two and three) in the algorithm. In step two, we introduce the local unlabeled pool to avoid selecting *out-of-scope* (we will define it later) examples. In step three, we tackle how to leverage the oracle answers in the hierarchy. We will discuss them in the following subsections.

[2] See http://www.dmoz.org/erz/ for DMOZ editing guidelines.

[3] On the root of hierarchy tree, every example is positive.

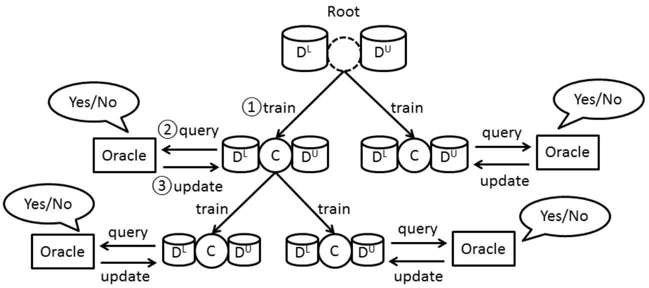


Fig. 1. The hierarchical active learning framework. The typical active learning steps are numbered 1, 2, 3 in the figure.

3.1 Unlabeled Pool Building Policy

From step one of our algorithm, we know that the training examples for a deep category (say c) must belong to its ancestor categories. However, it is likely that many unlabeled examples do not belong to the ancestor categories of c. We define those examples as *out-of-scope* examples. If those out-of-scope examples are selected by c, we may waste a lot of queries. Thus, instead of using one shared unlabeled pool [5] for all categories, we construct a local unlabeled pool on each of the categories. To filter out these out-of-scope examples, we use the predictions of the ancestor classifiers to build the local unlabeled pool. Specifically, given an unlabeled example x and a category c, only if all the ancestor classifiers of c predict x as positive, then we will place x into the local unlabeled pool of c.

3.2 Leveraging Oracle Answers

For the two answers ("Yes" or "No") from oracles, there are several possible ways to handle them. We give a brief overview here and discuss the detailed strategies in Section 5.

If the answer is "Yes", we can simply update the training set by directly including the queried example as a positive example. To better leverage the hierarchical relation, we can even add the positive example to all the ancestor categories. Furthermore, since the positive example is possibly a negative example on some of the sibling categories, we may consider including it as a negative example to the sibling categories.

If the answer is "No", we can not simply add the example as a negative example, since we don't know whether the queried example actually belongs to the ancestor categories. Thus, we could simply discard the example. Alternatively, we can also query the oracle on the parent category to see if the example belongs to the parent category, but the extra query may be wasted if the answer is "No".

In the following parts, we will first present our experimental configuration, and then empirically explore whether our framework can be effectively applied to hierarchical classification and whether different strategies described above can indeed improve active learning.

4 Experimental Configuration

4.1 Datasets

We utilize four real-world hierarchical text datasets (20 Newsgroups, OHSUMED, RCV1 and DMOZ) in our experiments. They are common benchmark datasets for evaluation of text classification methods. We give a brief introduction of the datasets. The statistic information of the four datasets is shown in Table 1.

Table 1. The statistic information of the four datasets. Cardinality is the average number of categories per example (i.e., multi-label datasets).

Dataset	Features	Examples	Categories	Levels	Cardinality
20 Newsgroups	61,188	18,774	27	3	2.202
OHSUMED	12,427	16,074	86	4	1.916
RCV1	47,236	23,049	96	4	3.182
DMOZ	92,262	12,735	91	3	2.464

The first dataset is *20 Newsgroups*[4]. It is a collection of newsgroup documents partitioned evenly across 20 different newsgroups. We group these categories based on subject matter into a three-level topic hierarchy which has 27 categories. The second dataset is *OHSUMED*[5]. It is a clinically-oriented MEDLINE dataset with a hierarchy of twelve levels. In our experiments, we only use the sub-hierarchy under subcategory "heart diseases" which is well-studied and usually taken as a benchmark dataset for text classification [8,13]. The third dataset is *RCV1* [9]. It includes three classification tasks: topic, industrial and regional classification. In our experiments, we focus on the topic classification task.[6] The last dataset is *DMOZ*. It is a human-edited web directory with webpages manually organized into a complex hierarchy. *DMOZ* is extracted from a sub collection rooted at "Science" and it has three-level category hierarchy.[7]

4.2 Performance Measure

To evaluate the performance in hierarchical classification, we adopt the hierarchical F-measure, which has been widely used in hierarchical classification for evaluation [17,3,14]. The definition the hierarchical F-measure is as follows,

$$hF = \frac{2 \times hP \times hR}{hP + hR} \quad where \quad hP = \frac{\sum_i |\hat{P}_i \bigcap \hat{T}_i|}{\sum_i |\hat{P}_i|} \quad hR = \frac{\sum_i |\hat{P}_i \bigcap \hat{T}_i|}{\sum_i |\hat{T}_i|} \tag{1}$$

[4] http://people.csail.mit.edu/jrennie/20Newsgroups/

[5] http://ir.ohsu.edu/ohsumed/

[6] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

[7] http://olc.ijs.si/dmozReadme.html

where hP and hR are the hierarchical precision and the hierarchical recall, $\hat{P}_i$ is the set consisting of the most specific categories predicted for test example i and all its (their) ancestor categories and $\hat{T}_i$ is the set consisting of the true most specific categories of test example i and all its (their) ancestor categories.[14]

4.3 Active Learning Setup

In our experiment, linear Support Vector Machine (SVM) is used as the base classifier on each category in the hierarchy, since the high dimensionality of text data usually results in the dataset being linearly separable [16]. Specifically, LIBLINEAR [7] package is used as the implementation of linear SVM. For LIBLINEAR, there are primarily two parameters C and W that will affect the performance. C is the penalty coefficient for training errors and W balances the penalty on the two classes. In our experiment, we set $C = 1000$ and W as the negative class proportion. For example, if the class ratio of positive and negative class in the training set is 1:9, then $W = 0.9$. The purpose is to give more penalty to the error on the minority class.

For active learning, due to the simplicity and effectiveness of *Uncertainty Sampling*[8], we adopt uncertainty sampling as the strategy to select the informative examples from the unlabeled pool. It should be noted that our hierarchical active learning framework is independent of the specific active learning strategy. Other strategies, such as expected error reduction [12] and representative sampling [18] can also be used. We will study them in the future.

We split all the four datasets into labeled (1%), unlabeled (89%) and testing (10%) parts. As we already know the labels of unlabeled examples, we will use the simulating oracles instead of the real human oracles (experts). We set a query limit (see Section 5.1). The training process is decomposed into a sequence of iterations. In each iteration, each category simultaneously selects a fixed number of examples[9] from its local unlabeled pool and queries the oracles (one query will be consumed when we ask one oracle for one label). After each category updates its training set, we recompute the parameter W and update the classification model. The entire training process terminates when the number of queries consumed exceeds the query limit. To reduce the randomness impact of the dataset split, we repeat this active learning process for 10 times. All the results (curves) in the following experiments are averaged over the 10 independent runs and accompanied by error bars indicating the 95% confidence interval.

5 Empirical Study

In this section, we will first experimentally study the standard version of our active learning framework for hierarchical text classification, then propose several improved versions and compare them with the previous version.

[8] Uncertain sampling in active learning selects the unlabeled example that is closest to the decision boundary of the classifier.

[9] We heuristically use logarithm of the unlabeled pool size to calculate the number of selected examples for each category.

5.1 Standard Hierarchical Active Learner

In order to validate our active learning framework, we will first compare its standard version (we call it standard hierarchical active learner) with the baseline learner. The standard hierarchical active learner uses intuitive strategies to handle oracle answers (see Section 3.2) in deep categories. If the oracle answer is “Yes”, the standard hierarchical active learner directly includes the example as a positive example; if “No”, it simply discards the example. On the other hand, the baseline learner is actually the non-active version of the standard hierarchical active learner. Instead of selecting the most informative examples, it selects unlabeled examples randomly on each category.

Empirical Comparison: We set the query limit as 50 × $|C|$ where $|C|$ is the total umber of categories in the hierarchy. Thus, in our experiments the query limits for the four datasets are 1,350, 4,300, 4,800 and 4,850 respectively. We denote the standard hierarchical active learner as AC and the baseline learner as RD. Figure 2 plots the average learning curves for AC and RD on the four datasets. As we can see, on all the datasets AC performs significantly better than RD. This result is reasonable since the unlabeled examples selected by AC are more informative than RD on all the categories in the hierarchy. From the curves, it is apparent that to achieve the best performance of RD, AC needs significantly fewer queries (approximately 43% to 82% queries can be saved)[10].

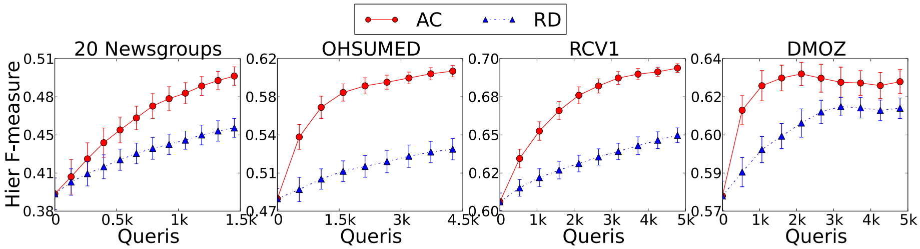


Fig. 2. Comparison between AC and RD in terms of the hierarchical F-measure. X axis is the number of queries consumed and Y axis is the hierarchical F-measure.

Although the standard hierarchical active learner (AC) significantly reduces the number of oracle queries compared to the baseline learner (RD), we should note that there is no interaction between categories in the hierarchy (e.g., each category independently selects examples and queries oracle). Our question is: can we further improve the performance of the standard active learner by taking into account the hierarchical relation of different categories? We will explore several leveraging strategies in the following subsections.

[10] In *20 Newsgroups*, RD uses 1,350 queries to achieve 0.46 in terms of the hierarchical F-measure, while AC only uses 750 queries. Thus, $(1350-750)/1350 = 44.4\%$ of the total queries are saved. The savings for other datasets are 82.5%, 72.9% and 43.3%.

5.2 Leveraging Positive Examples in Hierarchy

As mentioned in Section 3.2, when the oracle on a category answers "Yes" for an example, we can directly include the example into the training set on that category as a positive example. Furthermore, according to the category relation in a hierarchy, if an example belongs to a category, it will definitely belong to all the ancestor categories. Thus, we can propagate the example (as a positive example) to all its ancestor categories. In such cases, the ancestor classifiers can obtain *free* positive examples for training without any query. It coincides with the goal of active learning: reducing the human labeling cost!

Based on the intuition, we propose a new strategy *Propagate* to propagate the examples to the ancestor classifiers when the answer from oracle is "Yes". The basic idea is as follows. In each iteration of the active learning process, after we query an oracle for each selected example, if the answer from the oracle is "Yes", we propagate this example to the training sets of all the ancestor categories as positive. At the end of the iteration, each category combines all the propagated positive examples and the examples selected by itself to update its classifier.

Empirical Comparison: We integrate *Propagate* to the standard hierarchical active learner (we name the integrated version as *AC+*) and then compare it with the original *AC*. The first row of Figure 3 shows the learning curves of *AC+* and

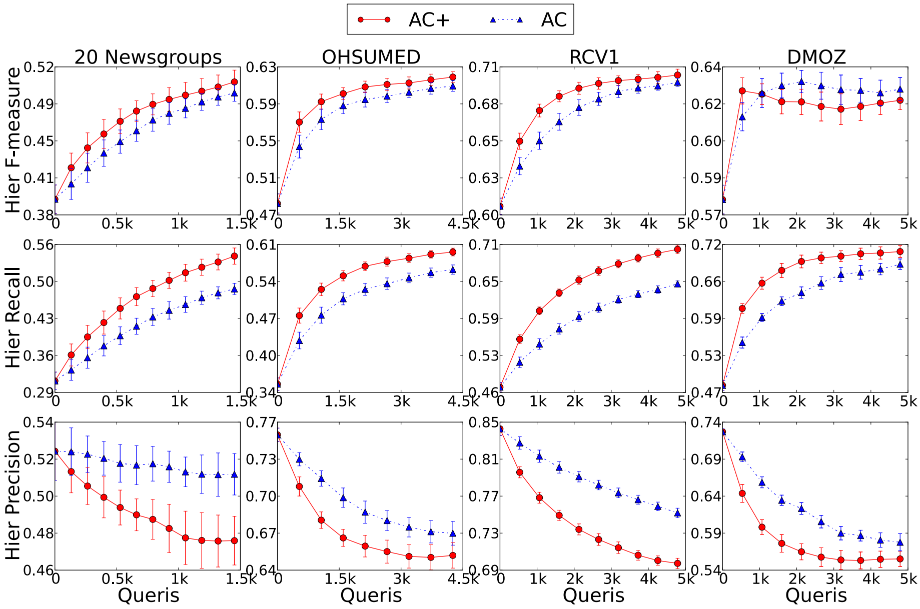


Fig. 3. Comparison between *AC+* and *AC* in terms of the hierarchical F-measure (first row), recall (second row) and precision (third row)

AC on the four datasets in terms of the hierarchical F-measure. Overall, the performance of *AC+* is slightly better than that of *AC*. By propagating positive examples, the top-level classifiers of *AC+* can receive a large number of positive examples and thus the (hierarchical) recall of *AC+* increases faster than *AC* as shown in the second row. This is the reason why *AC+* can defeat *AC* on the first three datasets. However, from the third row, we can see the hierarchical precision of *AC+* actually degrades very sharply since the class distribution of the training set has been altered by the propagated positive examples. It thus weakens the boosting effect in the hierarchical recall and hinders the improvement of overall performance in the hierarchical F-measure.

Since positive examples can benefit the hierarchical recall, can we leverage negative examples to help maintain the hierarchical precision so as to further improve *AC+*? We will propose two possible solutions in the following.

5.3 Leveraging Negative Examples in Hierarchy

We introduce two strategies to leverage negative examples. One is to query parent oracles when the oracle answers “No”; the other is to predict the negative labels for sibling categories when the oracle answers “Yes”.

Querying Negative Examples: For deep categories, when the oracle answers “No”, we actually discard the selected example in *AC+* (as well as in *AC*, see Section 5.1). However, in this case, the training set may miss a negative example and also possibly an informative example. Furthermore, if we keep throwing away those examples whenever oracle says “No”, the classifiers may not have chance to learn negative examples. On the other hand, if we include this example, we may introduce noise to the training set, since the example may not belong to the parent category, thus an out-of-scope example (see Section 3.1).

How can we deal with the two cases? We introduce a complementary strategy called *Query*. In fact, the parent oracle can help us decide between the two cases. We only need to issue another query to the parent oracle on whether this example belongs to it. If the answer from the parent oracle is “Yes”, we can safely include this example as a negative example to the current category. If the answer is “No”, we can directly discard it. Here, we do not need to further query all the ancestor oracles, since the example is already out of scope of the current category and thus can not be included into its training set. There is a trade-off. As one more query is asked, we may obtain an informative negative example, but we may also waste a query. Therefore, it is non-trivial if this strategy works or not.

Predicting Negative Labels: When the oracle on a category (say “Astronomy”) answers “Yes” for an example, it is very likely that this example may not belong to its sibling categories such as “Chemistry” and “Social Science”. In this case, can we add this example as a negative example to its sibling categories? In those datasets where each example only belongs to one single category path, we can safely do so. It is because for the categories under the same parent, the example can only belong to at most one category. However, in most of the hierarchical datasets, the example belongs to multiple paths. In this case, it may be positive on some sibling categories. If we include this example as negative to the sibling categories, we may introduce noise.

To decide which sibling categories an example can be included as negative, we adopt a conservative heuristic strategy called *Predict*. Basically, when a positive example is included into a category, we add this example as negative to those sibling categories that the example is least likely to belong to. Specifically, if we know a queried example x is positive on a category c, we choose m sibling categories with the minimum probabilities (estimated by Platts Calibration [11]). We set

$$m = n - \max_{x \in D_L} \Psi_{\uparrow c}(x), \tag{2}$$

where D_L is the labeled set, $\uparrow c$ is the parent category of c, n is the number of children categories of $\uparrow c$, $\Psi_{\uparrow c}(x)$ is the number of categories under $\uparrow c$ that the example x belongs to.

Empirical Comparison: We integrate the two strategies *Query* and *Predict* discussed above into *AC+* and then compare the two integrated versions (*AC+Q* and *AC+P*) with the original *AC+*. Since in *AC+* positive examples are propagated, we can use this feature to further boost *AC+Q* and *AC+P*. For *AC+Q*, when the parent oracle answers "Yes", besides obtaining a negative example, we can also propagate this example as a positive example to all the ancestor categories. For *AC+P*, as a positive example is propagated, we can actually apply *Predict* to all the ancestor categories.

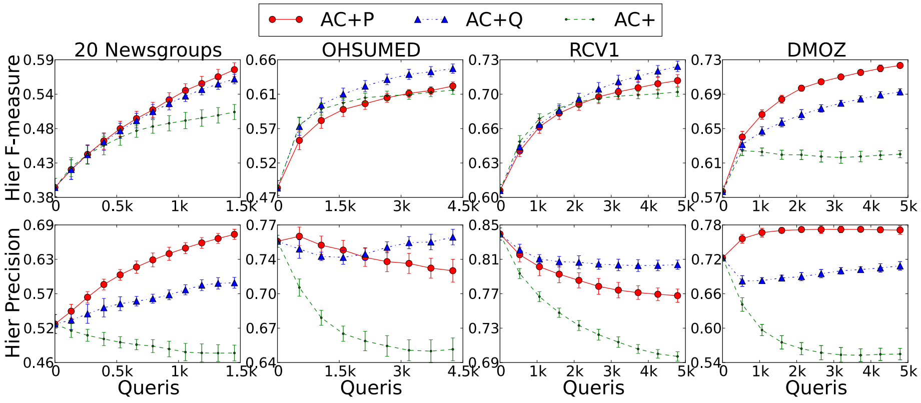


Fig. 4. Comparison between *AC+P*, *AC+Q* and *AC+* in terms of the hierarchical F-measure (upper row) and precision (bottom row)

We plot their learning curves for the hierarchical F-measure and the hierarchical precision on the four datasets in Figure 4. As we can see in the figure, both *AC+Q* and *AC+P* achieve better performance of the hierarchical F-measure than *AC+*. By introducing more negative examples, both methods maintain or even increase the hierarchical precision (see the bottom row of Figure 4). As we mentioned before, *AC+Q* may waste queries when the parent oracle answers "No". However, we discover that the average number of informative examples obtained per query for *AC+Q* is much larger than *AC+* (at least 0.2 higher per query). It means that it is actually worthwhile to issue another query in *AC+Q*. Another question is whether *AC+P* introduces noise to the training sets. According to our calculation, the noise rate is at most 5% on all the four datasets. Hence, it is reasonable that *AC+Q* and *AC+P* can further improve *AC+*.

However, between *AC+Q* and *AC+P*, there is no consistent winner on all the four datasets. On 20 Newsgroup and DMOZ, *AC+P* achieves higher performance, while on OHSUMED and RCV1, *AC+Q* is more promising. We also try to make a simple combination of *Query* and *Predict* with *AC+* (we call it *AC+QP*), but the performance is not significantly better than *AC+Q* and *AC+P*. We will explore a smarter way to combine them in our future work.

Finally, we compare the improved versions *AC+Q* and *AC+P* with the non-active version *RD*. We find that *AC+Q* and *AC+P* can save approximately 74% to 90% of the total queries. The savings for the four datasets are 74.1%, 88.4%, 83.3% and 90% respectively (these numbers are derived from Figures 2 and 4).

To summarize, we propose several improved versions (*AC+*, *AC+Q* and *AC+P*) in addition to the standard version (*AC*) of our hierarchical active learning framework. According to our empirical studies, we discover that in terms of the hierarchical F-measure, *AC+Q* and *AC+P* are significantly better than *AC+*, which in turn is slightly better than *AC*, which in turn outperforms *RD* significantly. In terms of query savings, our best versions *AC+Q* and *AC+P* need significantly fewer queries than the baseline learner *RD*.

6 Conclusion

We propose a new multi-oracle setting for active learning in hierarchical text classification as well as an effective active learning framework for this setting. We explore different solutions which attempt to utilize the hierarchical relation between categories to improve active learning. We also discover that propagating positive examples to the ancestor categories can improve the overall performance of hierarchical active learning. However, it also decreases the precision. To handle this problem, we propose two additional strategies to leverage negative examples in the hierarchy. Our empirical study shows both of them can further boost the performance. Our best strategy proposed can save a considerable number of queries (74% to 90%) compared to the baseline learner. In our future work, we will extend our hierarchical active learning algorithms with more advanced strategies to reduce queries further.

References

1. Ceci, M., Malerba, D.: Classifying web documents in a hierarchy of categories: a comprehensive study. J. Intell. Inf. Syst. 28, 37–78 (2007)
2. D'Alessio, S., Murray, K., Schiaffino, R., Kershenbaum, A.: The effect of using hierarchical classifiers in text categorization. In: RIAO 2000, pp. 302–313 (2000)
3. Daraselia, N., Yuryev, A., Egorov, S., Mazo, I., Ispolatov, I.: Automatic extraction of gene ontology annotation and its correlation with clusters in protein networks. BMC Bioinformatics 8(1), 243 (2007)
4. Donmez, P., Carbonell, J.G.: Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In: CIKM 2008, pp. 619–628 (2008)
5. Esuli, A., Sebastiani, F.: Active Learning Strategies for Multi-Label Text Classification. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) ECIR 2009. LNCS, vol. 5478, pp. 102–113. Springer, Heidelberg (2009)
6. Fagni, T., Sebastiani, F.: Selecting negative examples for hierarchical text classification: An experimental comparison. J. Am. Soc. Inf. Sci. Technol. 61, 2256–2265 (2010)
7. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. The Journal of Machine Learning Research 9, 1871–1874 (2008)
8. Lam, W., Ho, C.Y.: Using a generalized instance set for automatic text categorization. In: SIGIR 1998, pp. 81–89 (1998)
9. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: Rcv1: A new benchmark collection for text categorization research. J. Mach. Learn. Res. 5, 361–397 (2004)
10. Nowak, S., Rüger, S.: How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In: MIR 2010, pp. 557–566 (2010)
11. Platt, J.C.: Probabilistic outputs for support vector machines. In: Advances in Large Margin Classifiers, pp. 61–74 (1999)
12. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: ICML 2001, pp. 441–448 (2001)
13. Ruiz, M.E., Srinivasan, P.: Hierarchical neural networks for text categorization (poster abstract). In: SIGIR 1999, pp. 281–282 (1999)
14. Silla Jr., C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. Data Min. Knowl. Discov. 22, 31–72 (2011)
15. Sun, A., Lim, E.-P.: Hierarchical text classification and evaluation. In: ICDM 2001, pp. 521–528 (2001)
16. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. J. Mach. Learn. Res. 2, 45–66 (2002)
17. Verspoor, K., Cohn, J., Mniszewski, S., Joslyn, C.: Categorization approach to automated ontological function annotation. In: Protein Science, pp. 1544–1549 (2006)
18. Xu, Z., Yu, K., Tresp, V., Xu, X., Wang, J.: Representative Sampling for Text Classification Using Support Vector Machines. In: Sebastiani, F. (ed.) ECIR 2003. LNCS, vol. 2633, pp. 393–407. Springer, Heidelberg (2003)
19. Xue, G.R., Xing, D., Yang, Q., Yu, Y.: Deep classification in large-scale text hierarchies. In: SIGIR 2008, pp. 619–626 (2008)
20. Yang, B., Sun, J.T., Wang, T., Chen, Z.: Effective multi-label active learning for text classification. In: KDD 2009, pp. 917–926 (2009)

TeamSkill Evolved: Mixed Classification Schemes for Team-Based Multi-player Games

Colin DeLong and Jaideep Srivastava

Department of Computer Science,
University of Minnesota
{delong,srivasta}@cs.umn.edu
http://www.cs.umn.edu

Abstract. In this paper, we introduce several approaches for maintaining weights over the aggregate skill ratings of subgroups of teams during the skill assessment process and extend our earlier work in this area to include game-specific performance measures as features alongside aggregate skill ratings as part of the online prediction task. We find that the inclusion of these game-specific measures do not improve prediction accuracy in the general case, but *do* when competing teams are considered evenly matched. As such, we develop a "mixed" classification method called TeamSkill-EVMixed which selects a classifier based on a threshold determined by the prior probability of one team defeating another. This mixed classification method outperforms all previous approaches in most evaluation settings and particularly so in tournament environments. We also find that TeamSkill-EVMixed's ability to perform well in close games is especially useful early on in the rating process where little game history is available.

Keywords: Player rating systems, competitive gaming, perceptron, passive aggressive algorithm, confidence-weighted learning.

1 Introduction

In games, the challenge of ascertaining one player or team's advantage over their opponents continues to be an open research problem. In particular, the rise of online multi-player games has put the task of skill assessment front and center for game developers, wherein the long-term success or failure of a title is linked, in part, to the ability of players to find similarly-skilled teammates and opponents to play against. "Matchmaking", an automated process used to match players together for an online game, depends on accurate estimations of player skill at all times in order to reduce the likelihood of imbalanced matches. If one player or team is far superior to their opposition, the resulting game can frustrate less-skilled players and potentially lead to customer churn.

For games which focus on the online multi-player experience, including popular titles such as Halo, Call of Duty, and StarCraft 2, the task of appropriately matching up *millions* of players and teams of roughly equal skill is crucial - and

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 26–37, 2012.

daunting. With such large player populations, batch learning methods become impractical, neccesitating an online skill assessment process in which adjustments to a player's skill rating happen one game at a time, depending only on their existing rating and the outcome of the game. This task is made more difficult in titles centered around team-based competition, where interaction effects between teammates can be difficult to model and integrate into the assessment process.

Our work is concerned with this particular variant of the skill estimation problem. Although many approaches exist for skill estimation, such as the well-known Elo rating system [1] and the Glicko rating system [2], [3], they were primarily designed for one versus one competition settings (in games such as Chess or tennis) instead of team-based play. They can be altered to accomodate competitions involving teams, but, problematically, assume the performances of players in teams are independent from one another, thereby excluding potentially useful information regarding a team's collective "chemistry". More recent approaches [4] have explicitly modeled teams, but still assume player independence within teams, summing individual player ratings to produce an overall team rating.

"Team chemistry" is a widely-held notion in team sports [5] and is often cited as a key differentiating factor, particularly at the highest levels of competition. In the context of skill assessment in an online setting, however, less attention has been given to situations in which team chemistry would be expected to play a significant role, such as the case where the player population is highly-skilled individually, instead using data from a general population of players for evaluation [4].

Our previous work in this area [6] described several methods for capturing elements of "team chemistry" in the assessment process by maintaining skill ratings for subsets of teams as well as individuals, aggregating these ratings together for an overall team skill rating. One of the methods, TeamSkill-AllK-EV (hereafter referred to as EV), performed especially well in our evaluation. One drawback of EV, however, was that it weighted each aggregate n-sized subgroup skill rating uniformly in the final summation, leaving open the possibility that further improvements might be made through an adaptive weighting process.

In this paper, we build on our previous work by introducing five algorithms which address this drawback in various ways, TeamSkill-AllK-Ev-OL1 (OL1), TeamSkill-AllK-Ev-OL2 (OL2), TeamSkill-AllK-Ev-OL3 (OL3), TeamSkill-AllK-EVGen (EVGen), and TeamSkill-AllK-EVMixed (EVMixed). The first three - OL1, OL2, and OL3 - employ adaptive weighting frameworks to adjust the summation weights for each n-sized group skill rating and limit their feature set to data common across all team games: the players, team assignments, and the outcome of the game. For EVGen and EVMixed, however, we explore the use of EV's final prediction, the label of the winning team, as a feature to be included along with a set of game-specific performance metrics in a variety of online classification settings [7], [8], [9]. For EVMixed, a threshold based on EV's prior probability of one team defeating another is used to determine whether or not to include the metrics as features and, if not, the algorithm defers to

EV's predicted label. EVGen, in contrast, always includes the metrics during classification.

Evaluation is carried out on a carefully-compiled dataset consisting of tournament and scrimmage games between professional Halo 3 teams over the course of two years. Halo 3 is a first-person shooter (FPS) game which was played competitively in Major League Gaming (MLG), the largest professional video game league in the world, from 2008 through 2010. With MLG tournaments regularly featuring 250+ Halo teams vying for top placings, heavy emphasis is placed on teamwork, making this dataset ideal for the evaluation of interaction effects among teammates.

We find that EVMixed outperforms all other approaches in most cases, often by a significant margin. It performs particularly well in cases of limited game history and in "close" games where teams are almost evenly-matched. These results suggest that while game-specific features can play a role in skill assessment, their utility is limited to contexts in which the skill ratings of teams are similar. When they are not, the inclusion of game-specific information effectively adds noise to the dataset since their values aren't conditioned on the strength of their opponents.

The outline of this paper follows. Section 2 briefly describes some of the work related to the problem of skill assessment. In Section 3, we introduce our proposed approaches - OL1, OL2, OL3, EVGen, and EVMixed. In Section 4, we describe some of the key features of the dataset, our evaluation testbed, and share the results of our evaluation in terms of game outcome prediction accuracy. We then conclude with Section 5, discussing the results and future work.

2 Related Work

The foundations of modern skill assessment approaches date back to the work of Louis Leon Thurstone [10] who, in 1927, proposed the "law of comparitive judgement", a method by which the mean distance between two physical stimuli, such as perceived loudness, can be computed in terms of the standard deviation when the stimuli processes are normally-distributed. In 1952, Bradley-Terry-Luce (BTL) models [11] introduced a logistic variant of Thurstone's model, using taste preference measurements for evaluation. This work in turn led to the creation of the Elo rating system, introduced by Arpad Elo in 1959 [1], a professor and master chess player who sought to replace the US Chess Federation's Harkness rating system with one more theoretically sound. Similar to Thurstone, the Elo rating system assumes the process underlying each player's skill is normally-distributed with a constant skill variance parameter β^2 across all players, simplifying skill updates after each game.

However, this simplification was also Elo's biggest drawback since the "reliabilty" of a rating was unknown from player to player. To address this, the Glicko rating system [3], a Bayesian approach introduced in 1993 by Mark Glickman, allowed for player-specific skill variance, making it possible to determine the confidence in a player's rating over time and produce more conservative skill estimates.

With the release of the online gaming service Xbox Live in 2002, whose player population quickly grew into the millions, there was a need for a more generalized rating system incorporating the notion of teams as well as individual players. TrueSkill [4], published in 2006 by Ralf Herbrich and Thore Graepel, used a factor graph-based approach to meet this need. In TrueSkill, skill variance is also maintained for each player, but in contrast to Glicko, TrueSkill samples an expected performance given a player's skill rating which is then summed over all members of a team to produce an estimate of the collective skill of a team. Because the summation is over individual players, player performances are assumed to be independent from one another, leaving out potentially useful group-level interaction information. For team-based games in which highly-skilled players may coordinate their strategies, this lost interaction information can make the estimation of a team's advantage over another difficult, especially as players change teams.

Several other variants of the aforementioned approaches have also been introduced, including BTL models [12], [13], [14] and expectation propagation techniques for the analysis of paired comparison data [15].

3 Proposed Approaches

In our previous work [6], we sought to explicitly model group-level interaction effects during the skill assessment process, introducing four methods which took varying approaches to addressing this issue - TeamSkill-K, TeamSkill-AllK, TeamSkill-AllK-EV, and TeamSkill-AllK-LS. These approaches had in common the idea that ratings themselves need not be limited to individual players, but subsets of teams as well. Here, we modified the Elo, Glicko, and TrueSkill rating systems to be used as generic learners which maintained skill ratings for groups of players. In doing so, both group and player-level skill could be captured, producing a clearer picture of a team's collective skill. The key differences between these approaches was the amount of subgroup rating information used and the ways in which aggregate group skill ratings were weighted during the summation to produce a team's skill rating.

One of the approaches, EV, performed especially well during evaluation, improving on the unaltered versions of Glicko and TrueSkill, and, in most test cases, the other TeamSkill approaches as well. The main idea behind EV is to use all available group-level history, from groups of size $k = 1$ (individual players) to $k = K$ (the size of the team), and sum together the expected skill rating corresponding to each set of k-sized group ratings, weighting each uniformly:

$$s_i^* = \frac{K}{\sum_{k=1}^{K} (|h_i(k)| > 0)} \sum_{k=1}^{K} \frac{E[h_i(k)]}{k} \tag{3.1}$$

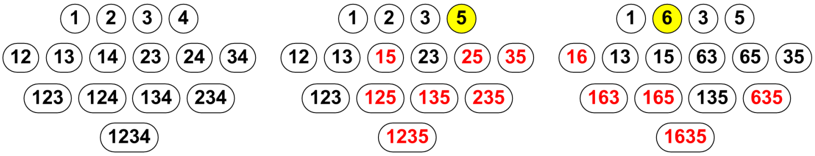


Fig. 1. The group history problem. This figure illustrates the group history available for a team of four players at three different time instances, proceeding chronologically from left to right. Black font indicates that history is available for a given group while red font indicates that history is not available.

In this notation, s_i^* is the estimated skill of team i and $h_i(k)$ is a function returning the set of skill ratings for player groups of size k in team i, including the empty set $\emptyset$ if none exist. When $h_i(k) \rightarrow \emptyset$, we let $E[h_i(k)] = 0$.

Despite its excellent results, EV is a "naive" approach, lacking a means of updating the summation weights, potentially leading to suboptimal performance. To that end, we introduce three adaptive frameworks which allow the summation weights to vary over time - TeamSkill-AllK-Ev-OL1 (OL1), TeamSkill-AllK-Ev-OL2 (OL2), and TeamSkill-AllK-Ev-OL3 (OL3).

3.1 TeamSkill-AllK-Ev-OL1

When attempting to construct an overall team skill rating, one key challenge to overcome is the fact that the amount of group history can vary over time. Consider figure 1: after the first game is played, history is available for all possible groups of players. Later, player 4 leaves the team and is replaced by player 5, who has never played with players 1, 2, or 3, leaving only a subset of history available and none for the team as a whole. Then in the final step, player 2 leaves and is replaced by player 6, who has played with player 3 and 5 before, but never both on the same team, resulting in yet another variant of the team's collective group-level history. The feature space is constantly expanding and contracting over time, making it difficult to know how best to combine the group-level ratings together. In OL1, we address this issue by maintaining a weight w_k for each aggregate group skill rating of size k, contracting $\mathbf{w}$ during summation by uniformly redistributing the weights from indicies in the weight vector not present in the available aggregate group skill rating history. Given the winning team i, w_k is updated by computing to what extent each of the aggregate rating's prior probability of team i defeating some team j according to TeamSkill-K [6], $P_k(i > j)$, is better than random, increasing the weight of w_k for a correctly-predicted outcome.

$$1 \leq \beta \leq \infty, w_k^0 = \frac{1}{K}, K' = \min\left(\max_{k \leq K}\left(|h_i(k)| > 0\right), \max_{k \leq K}\left(|h_j(k)| > 0\right)\right) \tag{3.2}$$

$$u = \frac{1}{K'} \sum_{k > K'} w_k^t \tag{3.3}$$

$$w'^t_{(k \leq K')} = w^t_{(k \leq K')} + u \tag{3.4}$$

$$s_i^* = \sum_{k=1}^{K'} w'^t_k E[h_i(k)] \tag{3.5}$$

$$w^{t+1}_{(k \leq K')} = w^t_{(k \leq K')} \beta^{\frac{1}{2} + P_k(i > j)} \tag{3.6}$$

$$w_k^{t+1} = \frac{w_k^{t+1}}{\sum_{l=1}^{K} w_l^{t+1}} \tag{3.7}$$

The main drawback of this approach is that the weight for $k = 1$ eventually dominates the weight vector as it is the element of group history present in every game and, therefore, the weight most frequently increased relative to the weights of $k > 1$. Given enough game history, this classifier will converge to exactly $k = 1$ - the classifier corresponding the an unmodified version of the general learner (Elo, Glicko, or TrueSkill) it employs.

3.2 TeamSkill-AllK-Ev-OL2

OL2 attempts to remedy this by maintaining a weight matrix corresponding to the lower triangular of a KxK grid, or one weight vector $\mathbf{w}$ for each of the K possible summation situations given a team's group-level game history. This ameliorates the issue of the $k = 1$ weight increasing faster relative to the weights of $k > 1$ since each row in the KxK grid pertains to a situation where the length of the non-zero row elements equals K' (as defined previously).

$$s_i^* = \sum_{k=1}^{K'} w^t_{(K',k)} E[h_i(k)] \tag{3.8}$$

$$w^{t+1}_{(K',k \leq K')} = w^t_{(K',k \leq K')} \beta^{\frac{1}{2} + P_k(i > j)} \tag{3.9}$$

$$w^{t+1}_{(K',k)} = \frac{w^{t+1}_{(K',k)}}{\sum_{l=1}^{K'} w^{t+1}_{(K',l)}} \tag{3.10}$$

3.3 TeamSkill-AllK-Ev-OL3

OL3 works similarly to OL1 in most respects, but instead uses a predefined window of the d most recent games in which k-sized group history was available to compute its updates. In this way, the weights "follow" the most confidently-correct aggregate skill ratings for each window d. In the following, let $L_{d,k}$ be the number of games in the window d in which, for some k, TeamSkill-K incorrectly predicted the outcome of a game.

$$s_i^* = \sum_{k=1}^{K'} w_k'^t E[h_i(k)] \tag{3.11}$$

$$w_{(k \leq K')}^{t+1} = w_{(k \leq K')}^{t} \beta^{\frac{1}{2}+(d-L_{d,k})/d} \tag{3.12}$$

$$w_k^{t+1} = \frac{w_k^{t+1}}{\sum_{l=1}^{K} w_l^{t+1}} \tag{3.13}$$

3.4 Using Game-Specific Data during Classification

OL1, OL2, and OL3 - like the other TeamSkill approaches - only use data available in all team-based games, namely the players, their team associations, and game outcome history. One natural question to ask is how well could we do if we included game-*specific* data during the step in which the label of the winning team is predicted. Though not ideal from a general implementation perspective, it is reasonable to assume that a carefully-chosen set of game-specific performance metrics might help produce a more accurate prediction. Here, we introduce two such methods - TeamSkill-AllK-EVGen (EVGen) and TeamSkill-AllK-EVMixed (EVMixed).

3.5 TeamSkill-AllK-EVGen

In EVGen, we create a feature set $\mathbf{x}_t$ from a combination of EV's predicted label $\{+1, -1\}$ of the winning team, $\hat{EV}_t$, and a set of n game-specific metrics $\mathbf{m}$. For Halo 3, several logical metrics are available, such as kill/death ratio and assist/death ratio (an assist is given to a player when they do more than half of the damage to a player who is eventually killed by another player), and act as rough measures of a team's in-game efficiency since players respawn after each death throughout the duration of a game. After compiling these metrics for each team, we take the difference between them for use in $\mathbf{x}_t$, adding in $\hat{EV}_t$ as the final feature. EV was chosen because of its superior performance in previous evaluations [6] as well as results from preliminary testing for this work, drawing from the pool of all previous approaches (including OL1, OL2, and OL3).

$$\mathbf{x}_t = (\hat{EV}_t, m_1, m_2, ..., m_n) \tag{3.14}$$

Having constructed the feature set $\mathbf{x}_t$, we use a more traditional online classification framework to predict the label of the winning team $\hat{y}_t$, such as the perceptron [7], online Passive-Aggressive algorithms [8], or Confidence-Weighted learning [9] (Note: substitute $\boldsymbol{\mu}_t$ for $\mathbf{w}_t$ in the latter):

$$\hat{y}_t = sign(\mathbf{w}_t \cdot \mathbf{x}_t) \tag{3.15}$$

After classification, the weight vector over the feature set is then updated according to the chosen learning framework.

3.6 TeamSkill-AllK-EVMixed

EVMixed introduces a slight variant to EVGen's overall strategy by selecting a classification approach based on whether or not both teams are considered relatively evenly-matched (that is, if a team's prior probability of winning according to EV, $P_{EV}^{t}(i > j)$, is close to 0.5). Here, if the prior probability of one team winning is within some ϵ of 0.5, we use the EVGen model for prediction. Otherwise we simply use EV's label. The approach is simple, as is the intuition behind it: if EV is sufficiently confident in its predicted label, then there is no need for additional feature information.

$$\hat{y}_t = \begin{cases} sign(\mathbf{w}_t \cdot \mathbf{x}_t) & \text{if } |P_{EV}^{t}(i > j) - 0.5| < \epsilon \\ \hat{EV}_t & \text{otherwise} \end{cases} \tag{3.16}$$

4 Evaluation

4.1 Dataset

We evaluate our proposed approaches using a dataset of 7,568 Halo 3 multiplayer games between professional teams. Each was played over the Internet on Microsoft's Xbox Live service in custom games (known as scrimmages) or on a local area network at an MLG tournament and includes information such as the players and teams competing, the date of the game, the map and game type, the result (win/loss) and score, and per-player statistics such as kills, deaths, and assists.

Characteristics unique to this dataset make it ideal for our evaluation purposes. First, it is common for players to change teams between tournaments, each of which is held roughly every 1-2 months, thereby allowing us to study the effects of "team chemistry" on performance without the assumption of degraded individual skill. Second, because every player is competing at such a high level, their individual skill isn't considered as important a factor in winning or losing a game as their ability to work together as a team.

4.2 Overall Results

The prediction accuracy of OL1, OL2, OL3, EVGen, and EVMixed were evaluated using a number of different subsets of the Halo 3 dataset:

- Games played in tournaments only, scrimmage games only, and both tournament and scrimmage games.
- All of the games, or just those games considered "close" (i.e., prior probability of one team winning close to 50%).

For comparison, we include results from the previous TeamSkill approaches as well. To compute the prior probability of t_1 defeating t_2, we use the negative CDF evaluated at 0 for the distribution corresponding to the difference between two independent, normally-distributed random variables (as in [6]). Games were labeled as "close" using a variant of the "challenge" method [4] in which the top

Table 1. Overall prediction accuracy for all test cases. **Bold cells** = highest accuracy; ***bolded/italicized*** = 2nd-highest accuracy.

Learner	Data	Close?	k=1	k=2	k=3	k=4	AllK	AlKEV	AlKLS	OL	OL2	OL3	EVGen	EVMxd
Elo	Both	N	0.645	0.642	0.636	0.631	0.642	0.645	0.633	0.645	0.645	***0.646***	0.574	**0.647**
		Y	0.512	0.494	0.497	0.485	0.493	0.5	0.489	0.495	0.495	0.502	**0.523**	***0.521***
	Tourn.	N	***0.639***	0.626	0.607	0.571	0.628	0.635	0.592	***0.639***	***0.639***	0.633	0.572	**0.643**
		Y	0.518	0.497	0.482	0.464	0.5	0.51	0.474	0.531	0.536	0.51	**0.549**	***0.544***
	Scrim.	N	***0.643***	0.639	0.639	0.631	0.642	0.64	0.633	***0.643***	***0.643***	0.64	0.583	**0.644**
		Y	0.503	0.487	0.492	0.476	0.496	0.488	0.476	0.499	0.498	0.487	**0.529**	***0.512***
Glicko	Both	N	0.636	0.63	0.632	0.635	***0.64***	***0.64***	0.633	0.637	0.637	***0.64***	0.581	**0.641**
		Y	0.522	0.564	0.562	0.547	0.569	0.57	0.548	0.524	0.552	***0.571***	0.528	**0.573**
	Tourn.	N	0.638	0.637	0.616	0.588	0.644	***0.647***	0.613	0.637	0.637	***0.647***	0.566	**0.657**
		Y	0.484	0.529	0.531	0.523	***0.576***	0.57	0.557	0.526	0.56	0.57	0.518	**0.62**
	Scrim.	N	0.631	0.635	0.637	0.637	**0.643**	0.637	0.634	0.635	0.636	0.637	0.582	***0.638***
		Y	0.496	0.559	***0.565***	0.522	***0.562***	0.551	0.524	0.531	0.551	0.551	0.525	0.554
TrueSkill	Both	N	0.635	0.641	0.636	0.63	0.638	0.642	0.632	0.635	0.636	**0.643**	0.572	**0.643**
		Y	0.516	0.555	0.542	0.542	0.552	0.56	0.548	0.536	0.544	**0.562**	0.522	***0.561***
	Tourn.	N	0.64	0.626	0.601	0.576	0.626	0.636	0.601	0.641	**0.644**	0.634	0.569	**0.653**
		Y	0.5	0.497	0.479	0.474	0.508	0.51	0.495	0.531	***0.547***	0.508	0.542	**0.573**
	Scrim.	N	0.636	**0.642**	0.639	0.632	0.636	0.638	0.634	0.636	0.637	0.637	0.581	***0.64***
		Y	0.504	**0.55**	0.542	0.53	0.541	0.54	0.533	0.548	**0.55**	0.543	0.522	0.542

20% closest games for one rating system are identified and presented to the other. Because we are interested in performance beyond that of unmodified general learners (i.e., $k = 1$), the closest games from $k = 1$ were presented to the other TeamSkill approaches while EV's closest games were presented to $k = 1$ (due to its evaluated performance in [6]). The following defaults were used for Elo ($\alpha = 0.07$, $\beta = 193.4364$, $\mu_0 = 1500$, $\sigma_0^2 = \beta^2$), Glicko ($q = log(10)/400$, $\mu_0 = 1500$, $\sigma_0^2 = 100^2$), and TrueSkill ($\epsilon = 0.5$, $\mu_0 = 25$, $\sigma_0^2 = (\mu_0/3)^2$, $\beta = \sigma_0^2/2$) according to [4] and [3]. For OL1/OL2, $\beta = 1.1$, OL3, $d = 20$. For EVGen/EVMixed ($\epsilon = 0.03$), the Passive-Aggressive II algorithm [8] was used for classification ($\alpha = 0.1$, $C = 0.001$, $\eta = 0.9$). The final feature set was comprised of cumulative and windowed (10 games of history) versions of team differences in average team and player-level kill/death ratio, assist/death ratio, kills/game, and assists/game.

From the results in table 1, it is clear that EVMixed performs the best overall, and in the widest array of evaluation conditions. It has the best performance in 10 of the 18 test cases and 16 of 18 in which it was at least second best, a testament to its consistency. EVGen's overall performance, however, is roughly 7-10% lower on average over all games, exceeding EVMixed's results only in 3 of the "close" game test cases.

4.3 Results over Time

Next we explore how these approaches perform over time by predicting the outcomes of games occuring prior to 10 tournaments which took place during 2008 and 2009, using tournament data only in order to isolate conditions in which we expect teamwork to be strongest. From figures 2 and 3, EVMixed's superior peformance is readily apparent. Of particular note, however, is how well EVMixed does when little history is available, having a roughly 64% accuracy just prior to the first tournament for all three learner cases. For close games,

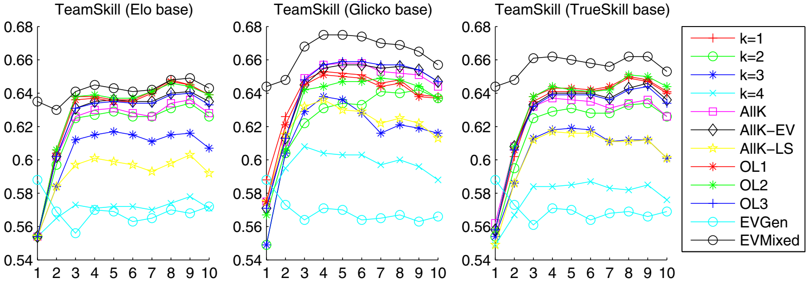


Fig. 2. Prediction accuracy over time for tournament games

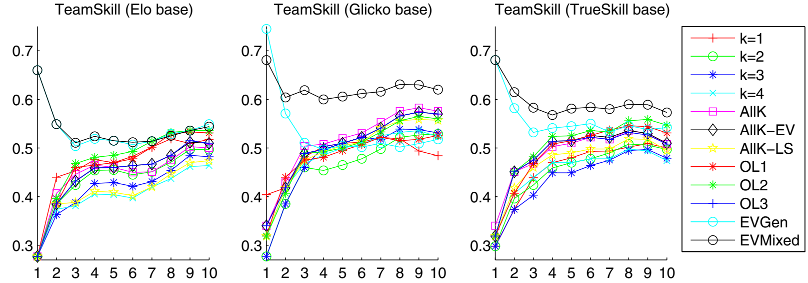


Fig. 3. Prediction accuracy over time for tournament games, close games only

both EVGen and EVMixed show strong results, eventually tapering off and approaching the other competing methods as more game history is observed.

4.4 Online Classification Variants

For EVGen and EVMixed, we investigated a number of different online classification frameworks - the perceptron [7], Passive-Aggressive algorithms [8], and Confidence-Weighted learning [9] - and evaluated them using a subset of the testbed from section 4.2. The results are shown in table 2. Though similar, the PA-II approach appears to be the most consistent overall (with CW-diag not far behind).

Table 2. Comparison of prediction accuracy by online classification framework using Glicko as the general learner

		EVGen			EVMixed		
Data	Close?	Perceptron	PA-II	CW-diag	Perceptron	PA-II	CW-diag
Both	N	0.575	***0.581***	**0.584**	**0.641**	**0.641**	**0.641**
	Y	***0.514***	**0.528**	**0.528**	**0.573**	**0.573**	**0.573**
Tourn.	N	0.543	**0.566**	***0.564***	***0.655***	**0.657**	**0.657**
	Y	0.474	**0.518**	***0.51***	0.609	**0.62**	***0.617***
Scrim.	N	0.575	***0.582***	**0.586**	**0.638**	**0.638**	***0.637***
	Y	***0.515***	**0.525**	0.512	**0.556**	***0.554***	0.551

5 Discussion

In sum, the results show EVMixed consistently outperforming competing approaches in a multitude of scenarios, often by great margins. Initially, we found the subpar performance of EVGen somewhat surprising given that the only difference between it and EVMixed is the classifier choice according to a given ϵ. Upon closer examination, the reason for this discrepency becomes clear: the game-specific data used to supplement the feature set was *not* weighted according to the strength of their opposition in each game, effectively adding "noise" in cases where the games were not considered close. Only the skill rating is a function of opposition skill, and as such, when the ratings of two teams are sufficiently divergent, the additional features are not necessary, nor desired. It follows that this is also the reason why both EVGen and EVMixed perform well in close games. Here, because the difference in skill ratings is small, the supplemental feature information tells us something about how two otherwise evenly-matched teams might perform if they competed. This is also why EVGen and EVMixed have excellent results when little game history has been observed - nearly all games are considered "close" early in the rating process.

Turning our attention back to OL1, OL2, and OL3, it's clear that little improvement was made relative to EV's results for any of these approaches. In fact, while the weights for OL1 eventually converge to the classifier $k = 1$, OL2's weights largely mimic EV's, suggesting there are more subtle group-level dynamics we need to pay attention to as this would only arise if the classifiers corresponding to $1 \leq k \leq K$ have somewhat similar ratings. OL3 also produces results similar to EV (even moreso than OL2), adding to the previous observation. While the results for OL1, OL2, and OL3 are unfortunate, the naive means by which EV weights each of the aggregated group-level skill ratings leaves the door open for improvement.

Our future work takes two directions. The first is to more fully explore what can be done to enhance the EVMixed model, perhaps by introducing a mechanism by which ϵ can vary over time or weighting player performances in-game by the strength of their opponents. The second is to derive an adaptive weighting framework which does improve on EV's results significantly, and then integrate it into EVGen and EVMixed.

6 Conclusions

In this paper, we extended our previous work by introducing three methods in which various strategies are used to maintain a set of weights over aggregate group-level skill rating information. Additionally, we explored the utility of incorporating game-specific data as features during the prediction process, describing two such approaches: EVGen and EVMixed. EVMixed outperformed all previous efforts in the vast majority of cases, leading to the conclusion that game-specific data is best included when teams are relatively evenly-matched, and disregarded otherwise.

Acknowledgments. We would like to thank members of the Data Mining Research Group for their feedback and suggestions. We would also like to thank Major League Gaming for making their 2008-2009 tournament data available.

References

1. Elo, A.: The Rating of Chess Players, Past and Present. Arco Publishing, New York (1978)
2. Glickman, M.: Paired Comparison Model with Time-Varying Parameters. PhD thesis. Harvard University, Cambridge, Massachusetts (1993)
3. Glickman, M.: Parameter estimation in large dynamic paired comparison experiments. Applied Statistics 48, 377–394 (1999)
4. Herbrich, R., Graepel, T.: Trueskill: A bayesian skill rating system. Microsoft Research, Tech. Rep. MSR-TR-2006-80 (2006)
5. Yukelson, D.: Principles of effective team building interventions in sport: A direct services approach at penn state university. Journal of Applied Sport Psychology 9(1), 73–96 (1997)
6. DeLong, C., Pathak, N., Erickson, K., Perrino, E., Shim, K., Srivastava, J.: TeamSkill: Modeling Team Chemistry in Online Multi-player Games. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part II. LNCS, vol. 6635, pp. 519–531. Springer, Heidelberg (2011)
7. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychological Review 65(6), 386–408 (1958)
8. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. Journal of Machine Learning Research 7, 551–585 (2006)
9. Crammer, K., Dredze, M., Pereira, F.: Exact convex confidence-weighted learning. In: Advances in Neural Information Processing Systems, vol. 21, pp. 345–352 (2009)
10. Thurstone, L.: Psychophysical analysis. American Journal of Psychology 38, 368–389 (1927)
11. Bradley, R.A., Terry, M.: Rank analysis of incomplete block designs: I. the method of paired comparisons. Biometrika 39(3/4), 324–345 (1952)
12. Coulom, R.: Whole-History Rating: A Bayesian Rating System for Players of Time-Varying Strength. In: van den Herik, H.J., Xu, X., Ma, Z., Winands, M.H.M. (eds.) CG 2008. LNCS, vol. 5131, pp. 113–124. Springer, Heidelberg (2008)
13. Huang, T., Lin, C., Weng, R.: Ranking individuals by group comparisons. Journal of Machine Learning Research 9, 2187–2216 (2008)
14. Menke, J.E., Reese, C.S., Martinez, T.R.: Hierarchical models for estimating individual ratings from group competitions. American Statistical Association (2007) (in preparation)
15. Birlutiu, A., Heskes, T.: Expectation Propagation for Rating Players in Sports Competitions. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 374–381. Springer, Heidelberg (2007)

A Novel Weighted Ensemble Technique for Time Series Forecasting

Ratnadip Adhikari and R.K. Agrawal

School of Computer and Systems Sciences,
Jawaharlal Nehru University, New Delhi-110067, India
{adhikari.ratan,rkajnu}@gmail.com

Abstract. Improvement of time series forecasting accuracy is an active research area having significant importance in many practical domains. Extensive works in literature suggest that substantial enhancement in accuracies can be achieved by combining forecasts from different models. However, forecasts combination is a difficult as well as a challenging task due to various reasons and often simple linear methods are used for this purpose. In this paper, we propose a nonlinear weighted ensemble mechanism for combining forecasts from multiple time series models. The proposed method considers the individual forecasts as well as the correlations in pairs of forecasts for creating the ensemble. A successive validation approach is formulated to determine the appropriate combination weights. Three popular models are used to build up the ensemble which is then empirically tested on three real-world time series. Obtained forecasting results, measured through three well-known error statistics demonstrate that the proposed ensemble method provides significantly better accuracies than each individual model.

Keywords: Time Series Forecasting, Ensemble Technique, Box-Jenkins Models, Artificial Neural Networks, Elman Networks.

1 Introduction

Time series forecasting has indispensable importance in many practical data mining applications. It is an ongoing dynamic area of research and over the years various forecasting models have been developed in literature [1,2]. A major concern in this regard is to improve the prediction accuracy of a model without sacrificing its flexibility, robustness, simplicity and efficiency. However, this is not at all an easy task and so far no single model alone can provide best forecasting results for all kinds of time series data [3,4].

Combining forecasts from conceptually different methods is a very effective way to improve the overall forecasting precisions. The earliest use of this practice started in 1969 with the monumental work of Bates and Granger [5]. Till then, numerous forecasts combination methods have been developed in literature [6,7,8]. The precious role of model combination in time series forecasting can be credited to the following facts: (a) by an adequate ensemble technique, the

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 38–49, 2012.

forecasting strengths of the participated models aggregate and their weaknesses diminish, thus enhancing the overall forecasting accuracy to a great extent, (b) often, there is a large uncertainty about the optimal forecasting model and in such situations combination strategies are the most appropriate alternatives to use, and (c) combining multiple forecasts can efficiently reduce errors arising from faulty assumptions, bias, or mistakes in the data [3].

The simple average is the most widely used forecasts combining technique. It is easy to understand, implement and interpret. However, this method is often criticized because it does not utilize the relative performances of the contributing models and is quite sensitive to the extreme errors [1,3]. As a result, other forms of averaging, e.g. trimmed mean, Winsorized mean, median, etc. have been studied in literature [9]. Another common method is the weighted linear combination of individual forecasts in which the weights are determined from the past forecast errors of the contributing models. But, this method completely ignores the possible relationships between two or more participating models and hence is not so adequate for combining nonstationary and chaotic data. Various modifications of this linear combination technique have also been suggested by researchers [9,10,11].

In this paper, we propose a weighted nonlinear framework for combining multiple time series models. Our approach is partially motivated by the work of Freitas and Rodrigues [12]. The proposed technique considers individual forecasts from different methods as well as the correlations between pairs of forecasts for combining. We consider three models, viz. Autoregressive Integrated Moving Average (ARIMA), Artificial Neural Network (ANN) and Elman ANN to build up the ensemble. An efficient methodology, based on a successive validation approach is formulated for finding the appropriate combination weights. The effectiveness of the proposed technique is tested on three real-world time series (one stationary and two nonstaionary financial data). The forecasting accuracies are evaluated in terms of the error measures: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Average Relative Variance (ARV).

The rest of the paper is organized as follows. Section 2 describes a number of common forecasts combination techniques. Our proposed ensemble scheme is presented in Sect. 3. In Sect. 4, we describe the three time series forecasting models, which are used here to build up the ensemble. Experimental results are reported in Sect. 5 and finally Sect. 6 concludes this paper.

2 Forecasts Combination Methods

Let the actual time series be $\mathbf{Y} = \{y_1, y_2, \dots, y_N\}$ and $\hat{\mathbf{Y}}^{(i)} = \{\hat{y}_1^{(i)}, \hat{y}_2^{(i)}, \dots, \hat{y}_N^{(i)}\}$ be its forecast obtained from the i^{th} method $(i = 1, 2, \dots, N)$. Then the series obtained from linearly combining these n forecasted series is given by:

$$\begin{cases} \hat{\mathbf{Y}}^{(\mathrm{c})} = \{\hat{y}_1^{(\mathrm{c})}, \hat{y}_2^{(\mathrm{c})}, \dots, \hat{y}_N^{(\mathrm{c})}\}, \\ \hat{y}_k^{(\mathrm{c})} = w_1 \hat{y}_k^{(1)} + w_2 \hat{y}_k^{(2)} + \dots + w_n \hat{y}_k^{(n)} = \sum\limits_{i=1}^{n} w_i \hat{y}_k^{(i)} \\ \forall k = 1, 2, \dots, N \end{cases} \tag{1}$$

where, w_i is the weight assigned to the i^{th} forecasting method. To ensure unbiasedness, sometimes it is assumed that the weights add up to unity. Different combination techniques are developed in literature which are based on different weight assignment schemes; some important among them are discussed here:

- In the *simple average*, all models are assigned equal weights, i.e. $w_i = 1/n\,(i = 1, 2, \ldots, n)$ [9,10].
- In the *trimmed average*, individual forecasts are combined by a simple arithmetic mean, excluding the worst performing $k\%$ of the models. Usually, the value of k is selected from the range of 10 to 30. This method is sensible only when $n \geq 3$ [9,10].
- In the *Winsorized average*, the i smallest and largest forecasts are selected and set to the $(i+1)^{\text{th}}$ smallest and largest forecasts, respectively [9].
- In the *error-based* combining, an individual weight is chosen to be inversely proportional to the past forecast error of the corresponding model [3].
- In the *outperformance* method, the weight assignments are based on the number of times a method performed best in the past [11].
- In the *variance-based* method, the optimal weights are determined by minimizing the total Sum of Squared Error (SSE) [7,10].

All the combination techniques, discussed above are linear in nature. The literature on nonlinear forecast combination methods is very limited and further research works are required in this area [10].

3 The Proposed Ensemble Technique

Our ensemble technique is an extension of the usual linear combination method in order to deal with the possible correlations between pairs of forecasts and is partially motivated from the work of Freitas and Rodrigues [12].

3.1 Mathematical Description

For simplicity, here we describe our ensemble technique for combining forecasts from three methods; but, it can be easily generalized . Let, the actual test dataset of a time series be $\mathbf{Y} = [y_1, y_2, \ldots, y_N]^{\mathrm{T}}$ with $\hat{\mathbf{Y}}^{(i)} = \left[\hat{y}_1^{(i)}, \hat{y}_2^{(i)}, \ldots, \hat{y}_N^{(i)}\right]^{\mathrm{T}}$ being its forecast obtained from the i^{th} method $(i = 1, 2, 3)$. Let, $\mu^{(i)}$ and $\sigma^{(i)}$ be the mean and standard deviation of $\hat{\mathbf{Y}}^{(i)}$ respectively. Then the combined forecast of $\mathbf{Y}$ is defined as: $\hat{\mathbf{Y}}^{(\mathrm{c})} = \left[\hat{y}_1^{(\mathrm{c})}, \hat{y}_2^{(\mathrm{c})}, \ldots, \hat{y}_N^{(\mathrm{c})}\right]^{\mathrm{T}}$, where

$$\begin{aligned}\hat{y}_k^{(\mathrm{c})} &= w_0 + w_1\hat{y}_k^{(1)} + w_2\hat{y}_k^{(2)} + w_3\hat{y}_k^{(3)}\\ &\quad + \theta_1 v_k^{(1)} v_k^{(2)} + \theta_2 v_k^{(2)} v_k^{(3)} + \theta_3 v_k^{(3)} v_k^{(1)}\end{aligned} \tag{2}$$

$$v_k^{(i)} = \left(y_k^{(i)} - \mu^{(i)}\right) / \left(\sigma^{(i)}\right)^2, \;\; \forall i = 1, 2, 3;\; k = 1, 2, \ldots, N.$$

In (2), the nonlinear terms are included in calculating $\hat{y}_k^{(\mathrm{c})}$ to take into account the correlation effects between two forecasts. It should be noted that for combining n methods, there will be $\binom{n}{2}$ nonlinear terms in (2).

3.2 Optimization of the Combination Weights

The combined forecast defined in (2) can be written in vector form as follows:

$$\hat{\mathbf{Y}}_k^{(\mathrm{c})} = \mathbf{F}\mathbf{w} + \mathbf{G}\boldsymbol{\theta} \tag{3}$$

where,

$$\begin{aligned}
\mathbf{w} &= [w_0, w_1, w_2, w_3]^{\mathrm{T}}, \ \boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]^{\mathrm{T}}. \\
\mathbf{F} &= \left[\mathbf{1} | \hat{\mathbf{Y}}^{(1)} | \hat{\mathbf{Y}}^{(2)} | \hat{\mathbf{Y}}^{(3)}\right]_{N \times 4}. \\
\mathbf{1} &= [1, 1, \ldots, 1]^{\mathrm{T}}. \\
\mathbf{G} &= \begin{bmatrix} v_1^{(1)} v_1^{(2)} & v_1^{(2)} v_1^{(3)} & v_1^{(3)} v_1^{(1)} \\ \vdots & \vdots & \vdots \\ v_N^{(1)} v_N^{(2)} & v_N^{(2)} v_N^{(3)} & v_N^{(3)} v_N^{(1)} \end{bmatrix}_{N \times 3}.
\end{aligned}$$

The weights are to be optimized by minimizing the forecast SSE, given by:

$$\begin{aligned}
\mathrm{SSE} &= \sum_{k=1}^{N} \left(y_k - \hat{y}_k^{(\mathrm{c})}\right)^2 \\
&= (\mathbf{Y} - \mathbf{F}\mathbf{w} - \mathbf{G}\boldsymbol{\theta})^{\mathrm{T}} (\mathbf{Y} - \mathbf{F}\mathbf{w} - \mathbf{G}\boldsymbol{\theta}) \\
&= \mathbf{Y}^{\mathrm{T}}\mathbf{Y} - 2\mathbf{w}^{\mathrm{T}}\mathbf{b} + \mathbf{w}^{\mathrm{T}}\mathbf{V}\mathbf{w} + \\
&\quad 2\mathbf{w}^{\mathrm{T}}\mathbf{Z}\boldsymbol{\theta} - 2\boldsymbol{\theta}^{\mathrm{T}}\mathbf{d} + \boldsymbol{\theta}^{\mathrm{T}}\mathbf{U}\boldsymbol{\theta}
\end{aligned} \tag{4}$$

where,

$$\begin{aligned}
\mathbf{V} &= \left[\mathbf{F}^{\mathrm{T}}\mathbf{F}\right]_{4 \times 4}, \mathbf{b} = \left[\mathbf{F}^{\mathrm{T}}\mathbf{Y}\right]_{4 \times 1}, \mathbf{Z} = \left[\mathbf{F}^{\mathrm{T}}\mathbf{G}\right]_{4 \times 3}, \\
\mathbf{d} &= \left[\mathbf{G}^{\mathrm{T}}\mathbf{Y}\right]_{3 \times 1}, \mathbf{U} = \left[\mathbf{G}^{\mathrm{T}}\mathbf{G}\right]_{3 \times 3}.
\end{aligned}$$

Now from $(\partial/\partial\mathbf{w})(\mathrm{SSE}) = 0$ and $(\partial/\partial\boldsymbol{\theta})(\mathrm{SSE}) = 0$, we get the following system of linear equations:

$$\begin{cases} \mathbf{V}\mathbf{w} + \mathbf{Z}\boldsymbol{\theta} = \mathbf{b} \\ \mathbf{Z}^{\mathrm{T}}\mathbf{w} + \mathbf{U}\boldsymbol{\theta} = \mathbf{d} \end{cases} \tag{5}$$

By solving (5), the optimal combination weights can be obtained as:

$$\begin{cases} \boldsymbol{\theta}_{\mathrm{opt}} = \left(\mathbf{U} - \mathbf{Z}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{Z}\right)^{-1} \left(\mathbf{d} - \mathbf{Z}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{b}\right) \\ \mathbf{w}_{\mathrm{opt}} = \mathbf{V}^{-1}\left(\mathbf{b} - \mathbf{Z}\boldsymbol{\theta}_{\mathrm{opt}}\right) \end{cases} \tag{6}$$

These optimal weights are determinable if and only if all the matrix inverses, involved in (6) are well-defined.

3.3 Approach for Weights Determination

The optimal weights in the proposed ensemble technique solely depend on the knowledge of the forecast SSE value. But, in practical applications it is unknown in advance, since the dataset Y to be forecasted is unknown. Due to this reason, we suggest a robust mechanism for estimating the combination weights from the training data. Here, we divide the available time series into a suitable number of pairs of training and validation subsets and determine the optimal weights for each pairs; the desired combination weights are then calculated as the mean of all these pairwise optimal weights. In this way, the past forecasting performances of the participating models are effectively utilized for weights determination.

The necessary steps of our ensemble scheme are outlined in Alg. 1.

Algorithm 1. Weighted nonlinear ensemble of multiple forecasts

```
Inputs: The training data: $\mathbf{Y} = [y_1, y_2, \ldots, y_N]^{\mathrm{T}}$ of the associated time series and its
    $n$ forecasts, obtained from the models: $\mathrm{M}_i \ (i = 1, 2, \ldots, n)$.
Output: The combined forecast vector $\hat{\mathbf{Y}}^{(\mathrm{c})} = \left[\hat{y}_1^{(\mathrm{c})}, \hat{y}_2^{(\mathrm{c})}, \ldots, \hat{y}_N^{(\mathrm{c})}\right]^{\mathrm{T}}$.
Steps:
 1. Select $base\_size$, $validation\_window$ and the positive integer $k$, such that:
        $base\_size + k \times validation\_window = N.$
 2. Set $j \leftarrow 1$.
 3. $\mathbf{W} \leftarrow$ empty, $\boldsymbol{\Theta} \leftarrow$ empty
    // initially set both the final weight matrices $\mathbf{W}$ and $\boldsymbol{\Theta}$ as the empty matrices of
    orders $n \times 1$ and $\binom{n}{2} \times 1$, respectively.
 4. while $j \leq k$ do
 5.    Define:
           $\alpha = base\_size + (j-1) \times validation\_window.$
           $\mathbf{Y}_{\mathrm{train}} = [y_1, y_2, \ldots, y_\alpha]^{\mathrm{T}}$.
           $\mathbf{Y}_{\mathrm{validation}} = [y_{\alpha+1}, y_{\alpha+2}, \ldots, y_{\alpha+validation\_window}]^{\mathrm{T}}$.
           // this step selects a pair of training and validation subsets of $\mathbf{Y}$.
 6.    Train each model $\mathrm{M}_i \ (i = 1, 2, \ldots, n)$ on $\mathbf{Y}_{\mathrm{train}}$ to forecast the corresponding
       $\mathbf{Y}_{\mathrm{validation}}$ dataset.
 7.    Determine the optimal combination weight vectors $\mathbf{w}_k$ and $\boldsymbol{\theta}_k$ using (6).
 8.    $\mathbf{W} = [\mathbf{W}|\mathbf{w}_k]$, $\boldsymbol{\Theta} = [\boldsymbol{\Theta}|\boldsymbol{\theta}_k]$// augment the currently found weight vectors to the
       corresponding weight matrices.
 9.    $j \leftarrow j + 1$.
10. end while
11. Calculate the final weight vectors $\mathbf{w}_{\mathrm{comb}}$ and $\boldsymbol{\theta}_{\mathrm{comb}}$ as:
        $\mathbf{w}_{\mathrm{comb}} = \mathrm{mean}\,(\mathbf{W}, \text{row-wise})$.
        $\boldsymbol{\theta}_{\mathrm{comb}} = \mathrm{mean}\,(\boldsymbol{\Theta}, \text{row-wise})$.
12. Use $\mathbf{w}_{\mathrm{comb}}$ and $\boldsymbol{\theta}_{\mathrm{comb}}$ to calculate the combined forecast vector according to (3).
```

4 Three Time Series Forecasting Models

In this paper, we consider three popular time series forecasting methods to build up our proposed ensemble. These three methods are briefly described here.

4.1 Autoregressive Integrated Moving Average (ARIMA)

The ARIMA models are the most widely used methods for time series forecasting, which are developed by Box and Jenkins in 1970 [2]. These models are based on the assumption that the successive observations of a time series are linearly generated from the past values and a random noise process. Mathematically, an ARIMA(p, d, q) model is represented as follows:

$$\phi(L)(1-L)^d y_t = \theta(L)\epsilon_t \ . \tag{7}$$

where,

$$\phi(L) = 1 - \sum_{i=1}^{p} \phi_i L^i,\ \theta(L) = 1 + \sum_{j=1}^{q} \theta_j L^j, \text{ and } L y_t = y_{t-1} \ .$$

The terms p, d, q are the model orders, which respectively refer to the *autoregressive*, *degree of differencing* and *moving average processes*; y_t is the actual time series and ϵ_t is a white noise process. In this model, a nonstationary time series is transformed to a stationary one by successively (d times) differencing it [2,4]. A single differencing is often sufficient for practical applications. The ARIMA$(0, 1, 0)$, i.e. $y_t - y_{t-1} = \epsilon_t$ is the popular Random Walk (RW) model which is frequently used in forecasting financial and stock-market data [4].

4.2 Artificial Neural Networks (ANNs)

ANNs are the most efficient computational intelligence models for time series forecasting [10]. Their outstanding characteristic is the nonlinear, nonparametric, data-driven and self-adaptive nature [4,13]. The *Multilayer Perceptrons (MLPs)* are the most popular ANN architectures in time series forecasting. MLPs are characterized by a feedforward network of an input layer, one or more hidden layers and an output layer, as depicted in Fig. 1. Each layer contains a number of nodes which are connected to those in the immediate next layer by acyclic links. In practical applications, usually a single hidden layer is used [4,10,13].

The notation (p, h, q) is commonly used to refer an ANN with p input, h hidden and q output nodes. The forecasting performance of an ANN model depends on a number of factors, e.g. the selection of a proper network architecture, training algorithm, activation functions, significant time lags, etc. However, no rigorous theoretical framework is available in this regard and often some experimental guidelines are followed [13]. In this paper, we use popular model selection criteria, e.g. *Akaike Information Criterion (AIC)* and *Bayesian Information Criterion (BIC)* [13,14] for selecting suitable ANN structures. The *Resilient Propagation*

(RP) [15,16] is applied as the network training algorithm and the logistic and identity functions are used as the hidden and output layer activation functions, respectively.

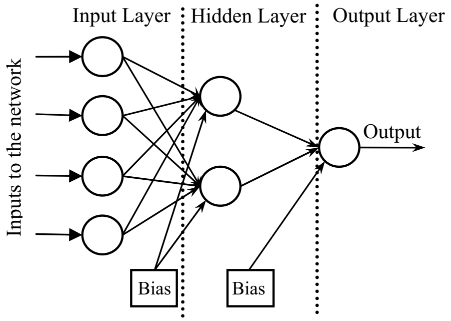


Fig. 1. Example of a multilayer feedforward ANN

4.3 Elman Artificial Neural Networks (EANNs)

Elman networks belong to the class of recurrent neural networks in which one extra layer, known as the context layer is introduced to recognize the spatial and temporal patterns in the input data [17]. The Elman networks contain two types of connections: feedforward and feedback. At every step, the outputs of the hidden layer are again fed back to the context layer, as shown in Fig. 2. This recurrence makes the network dynamic, so that it can perform non linear time-varying mappings of the associated nodes [16,17]. Unlike MLPs, there seems to be no general model selection guidelines in literature for the Elman ANNs [10]. However, it is a well-known fact that EANNs require much more hidden nodes than the simple feedforward ANNs in order to adequately model the temporal relationships [10,16]. In this paper, we use 24 hidden nodes and the training algorithm *traingdx* [16] for fitting EANNs.

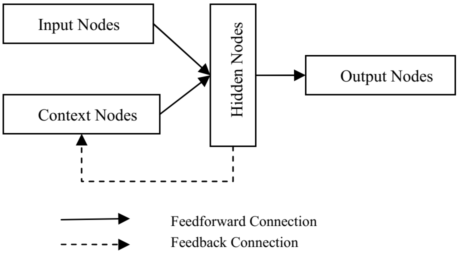


Fig. 2. Architecture of an Elman network

5 Experiments and Discussions

To empirically examine the performances of our proposed ensemble technique, three important real-world time series are used in this paper. These are the Wolfs sunspots, the daily closing price of S & P 500 index and the exchange rates between US Dollar (USD) and Indian Rupee (INR) time series. These time series are obtained from the Time Series Data Library (TSDL) [18], the Yahoo! Finance [19] and the Pacific FX database [20], respectively and are described in Table 1. The natural logarithms of the S & P data are used in our analysis. All three time series are divided into suitable training and testing sets. The training sets are used for fitting the three forecasting models as well as to build up the proposed ensemble; the testing sets are used to evaluate the out-of-sample forecasting performances of the fitted models and the ensemble.

The experiments in this paper are performed using MATLAB. For fitting ANN and EANN models, the neural network toolbox [16] is used. Forecasting efficacies of the models are evaluated through three well-known error statistics, viz. Mean Absolute Error (MAE), Mean Squared Error (MSE), and Average Relative Variance (ARV), which are defined below:

$$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n} |y_t - \hat{y}_t| \, , \; \mathrm{MSE} = \frac{1}{n}\sum_{t=1}^{n} (y_t - \hat{y}_t)^2 \, ,$$

$$\mathrm{ARV} = \left(\sum_{t=1}^{n} (y_t - \hat{y}_t)^2\right) \Big/ \left(\sum_{t=1}^{n} (\mu - \hat{y}_t)^2\right) ,$$

where, y_t and $\hat{y}_t$ are the actual and forecasted observations, respectively; N is the size and μ is the mean of the test set. For an efficient forecasting model, the values of these error measures are expected to be as less as possible.

The sunspots series is stationary with an approximate cycle of 11 years, as can be seen from Fig. 3(a). Following Zhang [4], the ARIMA$(9, 0, 0)$ (i.e. AR(9))

Table 1. Description of the time series datasets

Time Series	Description	Size
Sunspots	The annual number of observed sunspots (1700–1987).	Total size: 288 Training: 171 Testing: 67
S & P 500	Daily closing price of S & P 500 index (2 Jan. 2004–31 Dec. 2007)	Total size: 1006 Training: 800 Testing: 206
Exchange Rate	USD to INR exchange rates (1 July 2009–16 Sept. 2011)	Total size: 681 Training: 565 Testing: 116

and the $(7, 5, 1)$ ANN models are fitted to this time series. The EANN model is fitted with same numbers of input and output nodes as the ANN, but with 24 hidden nodes. For combining, we take $base\_size = 41$, $validation\_window = 20$ and the number of iterations $k = 9$.

The S & P and exchange rate are nonstationary financial series and both exhibit quite irregular patterns which can be observed from their respective time plots in Fig. 4(a) and Fig. 5(a). The RW-ARIMA model is most suitable for these type of time series[1]. For ANN modeling, the $(8, 6, 1)$ and $(6, 6, 1)$ network structures are used for S & P and exchange rate, respectively. As usual, the fitted EANN models have the same numbers of input and output nodes as the corresponding ANN models, but 24 hidden nodes. For combining, we take $base\_size = 200$, $validation\_window = 50$, $k = 12$ for the S & P data and $base\_size = 165$, $validation\_window = 40$, $k = 10$ for the exchange rate data.

In Table 2, we present the forecasting performances of ARIMA, ANN, EANN, simple average and the proposed ensemble scheme for all three time series.

Table 2. Forecasting results for the three time series

Error Measures		ARIMA	ANN	EANN	Average	Ensemble
Sunspots	MAE	17.63	15.58	14.71	13.59	12.50
	MSE	483.5	494.9	492.7	384.3	274.7
	ARV	0.216	0.308	0.280	0.218	0.120
S & P 500[2]	MAE	12.68	12.33	13.27	10.43	9.368
	MSE	28.27	21.58	24.61	15.69	13.59
	ARV	0.344	0.378	0.397	0.271	0.230
Exchange Rate	MAE	0.255	0.140	0.137	0.134	0.133
	MSE	0.105	0.032	0.030	0.029	0.028
	ARV	0.188	0.070	0.064	0.064	0.053

From Table 2, it can be seen that our ensemble technique has provided lowest forecast errors among all methods. Moreover, the proposed technique has also achieved considerably better forecasting accuracies than the simple average combination method, for all three time series. However, we have empirically observed that like the simple average, the performance of our ensemble method is also quite sensitive to the extreme errors of the component models.

In this paper, we use the term *Forecast Diagram* to refer the graph which shows the actual and forecasted observations of a time series. In each forecast diagram, the solid and dotted line respectively represents the test and forecasted time series. The forecast diagrams, obtained through our proposed ensemble method for sunspots, S & P and exchange rate series are depicted in Fig. 3(b), Fig. 4(b) and Fig. 5(b), respectively.

[1] In RW-ARIMA, the preceding observation is the best guide for the next prediction.

[2] Original MAE=Obtained MAE$\times 10^{-3}$; Original MSE=Obtained MSE$\times 10^{-5}$.

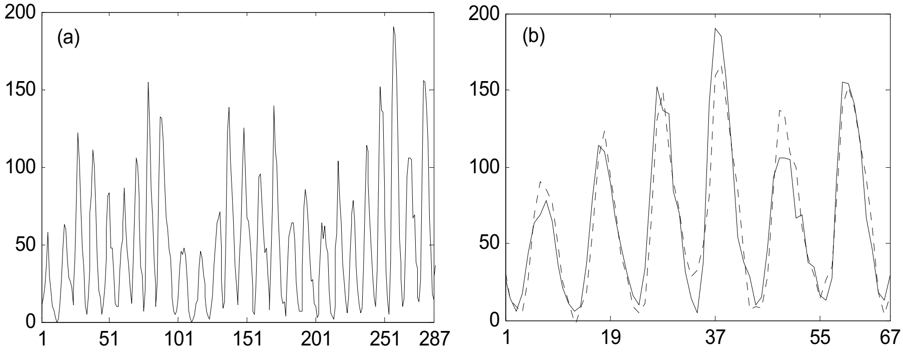


Fig. 3. (a) The sunspots series, (b) Ensemble forecast diagram for the sunspot series

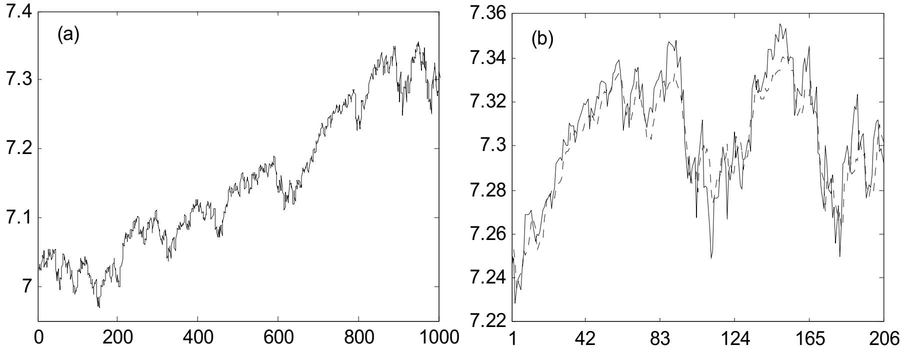


Fig. 4. (a) The S & P series, (b) Ensemble forecast diagram for the S & P series

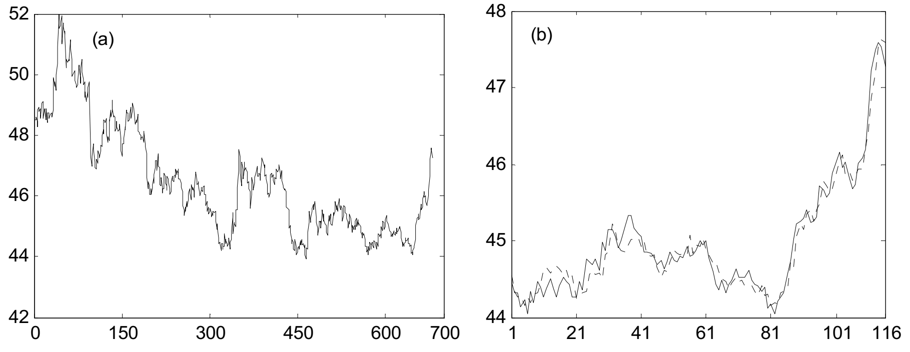


Fig. 5. (a) Exchange rate series, (b) Ensemble forecast diagram for exchange rate series

6 Conclusions

Improving the accuracy of time series forecasting is a major area of concern in many practical applications. Although numerous forecasting methods have been developed during the past few decades, but it is often quite difficult to select the best among them. It has been observed by many researchers that combining multiple forecasts effectively reduces the prediction errors and hence provides considerably increased accuracy.

In this paper, we propose a novel nonlinear weighted ensemble technique for forecasts combination. It is an extension of the common linear combination scheme in order to include possible correlation effects between the participating forecasts. An efficient successive validation mechanism is suggested for determining the appropriate combination weights. The empirical results with three real-world time series and three forecasting methods demonstrate that our proposed technique significantly outperforms each individual method in terms of obtained forecast accuracies. Moreover, it also provides considerably better results than the classic simple average combining technique. In future works, our ensemble mechanism can be further explored with other diverse forecasting models as well as other varieties of time series data.

Acknowledgments. The first author sincerely expresses his profound gratitude to the Council of Scientific and Industrial Research (CSIR) for the obtained financial support, which helped a lot in performing this research work.

References

1. Gooijer, J.G., Hyndman, R.J.: 25 Years of time series forecasting. J. Forecasting 22(3), 443–473 (2006)
2. Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control, 3rd edn. Holden-Day, California (1970)
3. Armstrong, J.S.: Combining Forecasts. In: Armstrong, J.S. (ed.) Principles of Forecasting: A Handbook for Researchers and Practitioners. Kluwer Academic Publishers, Norwell (2001)
4. Zhang, G.P.: Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50, 159–175 (2003)
5. Bates, J.M., Granger, C.W.J.: Combination of forecasts. Operational Research Quarterly 20(4), 451–468 (1969)
6. Clemen, R.T.: Combining forecasts: A review and annotated bibliography. J. Forecasting 5(4), 559–583 (1989)
7. Aksu, C., Gunter, S.: An empirical analysis of the accuracy of SA, OLS, ERLS and NRLS combination forecasts. J. Forecasting 8(1), 27–43 (1992)
8. Zou, H., Yang, Y.: Combining time series models for forecasting. J. Forecasting 20(1), 69–84 (2004)
9. Jose, V.R.R., Winkler, R.L.: Simple robust averages of forecasts: Some empirical results. International Journal of Forecasting 24(1), 163–169 (2008)
10. Lemke, C., Gabrys, B.: Meta-learning for time series forecasting and forecast combination. Neurocomputing 73, 2006–2016 (2010)

11. Bunn, D.: A Bayesian approach to the linear combination of forecasts. Operational Research Quarterly 26(2), 325–329 (1975)
12. Frietas, P.S., Rodrigues, A.J.: Model combination in neural-based forecasting. European Journal of Operational Research 173(3), 801–814 (2006)
13. Zhang, G., Patuwo, B.E., Hu, M.Y.: Forecasting with articial neural networks: The state of the art. J. Forecasting 14, 35–62 (1998)
14. Faraway, J., Chatfield, C.: Time series forecasting with neural networks: a comparative study using the airline data. J. Applied Statistics 47(2), 231–250 (1998)
15. Reidmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: The rprop algorithm. In: Proceedings of the IEEE Int. Conference on Neural Networks (ICNN), San Francisco, pp. 586–591 (1993)
16. Demuth, M., Beale, M., Hagan, M.: Neural Network Toolbox User's Guide. The MathWorks, Natic (2010)
17. Lim, C.P., Goh, W.Y.: The application of an ensemble of boosted Elman networks to time series prediction: A benchmark study. J. of Computational Intelligence 3, 119–126 (2005)
18. Time Series Data Library, http://robjhyndman.com/TSDL
19. Yahoo! Finance, http://finance.yahoo.com
20. Pacific FX database, http://fx.sauder.ubc.ca/data.html

Techniques for Efficient Learning without Search

Houssam Salem, Pramuditha Suraweera, Geoffrey I. Webb,
and Janice R. Boughton

Faculty of Information Technology, Monash University, VIC 3800, Australia
{Houssam.Salem,Pramuditha.Suraweera,Geoff.Webb}@monash.edu

Abstract. Averaged n-Dependence Estimators (AnDE) is a family of learning algorithms that range from low variance coupled with high bias through to high variance coupled with low bias. The asymptotic error of the lowest bias variant is the Bayes optimal. The AnDE family of algorithms have a training time that is linear with respect to the training examples, learn in a single pass through the data, support incremental learning, handle missing values directly and are robust in the face of noise. These characteristics make the algorithms particularly well suited to learning from large data. However, for higher orders of n they are very computationally demanding. This paper presents data structures and algorithms developed to reduce both memory and time for training and classification. These enhancements have enabled the evaluation and comparison of A3DE's effectiveness. The results provide further support for the hypothesis that as the number of training examples increases, decreasing error will be attained by members of the AnDE family with increasing levels of n.

Keywords: naive Bayes, semi-naive Bayes, probabilistic prediction.

1 Introduction

The classical classification learning paradigm performs search through a hypothesis space to identify a hypothesis that optimizes some objective function with respect to training data. Averaged n-Dependence Estimators (AnDE) [10] is an approach to learning without search or hypothesis selection, which represents a fundamental alternative to the classical learning paradigm.

The new paradigm gives rise to a family of algorithms, of which, Webb *et. al.* [10] hypothesize, the different members are suited for differing quantities of data. The algorithms range from low variance with high bias through to high variance with low bias. Webb *et. al.* suggest that members with low variance are suited for small datasets whereas members with low bias are suitable for large datasets. They claim that the asymptotic error of the lowest bias variant is Bayes optimal.

The algorithms in the family possess a unique set of features that are suitable for many applications. In particular, they have a training time that is linear with respect to the number of examples and can learn in a single pass through the training data without any need to maintain the training data in memory. Thus, they show great potential for very accurate classification from large data.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 50–61, 2012.

Further, they have direct capacity for incremental and anytime [6] learning, are robust in the face of noise and directly handle missing values. Importantly, evaluations have shown that their classification accuracy is competitive with the state-of-the-art in machine learning [10].

AnDE extends the underlying strategy of Averaged One-Dependence Estimators (AODE) [9], which relaxes the Naive Bayes (NB) independence assumption while retaining many of Naive Bayes's desirable computational and theoretical properties. The third member of the AnDE family, A2DE, has been shown to produce strong predictive accuracy over a wide range of data sets [10].

Although evaluations to date support the hypothesis that the predictive accuracy of AnDE increases for larger datasets with higher orders of n, the expected increase in accuracy comes at the cost of increased computational requirements. The current implementations further complicate the matter due to their inefficiencies. Thus, efficient implementation is critical. Except in cases of lower dimensional data, the computational requirements defeat a straightforward extension of Weka's AODE [11] to handle A3DE.

This paper presents data structures and algorithms that reduce both memory and time required for both training and classification. These improvements have enabled us to evaluate the effectiveness of A3DE on large datasets. The results provide further evidence that members of the AnDE family with increasing n are increasingly effective at classifying datasets of increasing size.

The remainder of the paper starts by introducing the AnDE family of algorithms. Section 3 outlines the memory representation developed to reduce memory usage. The enhancements made to reduce testing times are outlined in Section 4. Section 5 presents the results of evaluating the effectiveness of the enhancements. It also compares the effectiveness of A3DE with AnDE members with lower n. Finally, conclusions are outlined.

2 The AnDE Family of Algorithms

The classification problem can be stated as estimating, from a training sample τ of classified objects, the probability $\mathrm{P}(y \mid \mathbf{x})$ that an example $\mathbf{x} = \langle x_1, \ldots, x_a \rangle$ belongs to class y, where x_i is the value of the i^{th} attribute and $y \in c_1, \ldots, c_k$ that are k classes. As $\mathrm{P}(y \mid \mathbf{x}) \propto \mathrm{P}(y, \mathbf{x})$, we only need to estimate the latter.

The naive Bayes (NB) algorithm extrapolates to $\hat{\mathrm{P}}(\mathbf{x}, y)$ from each two dimensional probability estimate $\hat{\mathrm{P}}(x_i \mid y)$, by assuming that attributes are independent given the class. Based on this assumption,

$$\mathrm{P}(\mathbf{x} \mid y) = \prod_{i=1}^{a} \mathrm{P}(x_i \mid y). \tag{1}$$

Hence we classify using

$$\hat{\mathrm{P}}_{\mathrm{NB}}(y, \mathbf{x}) = \hat{\mathrm{P}}(y) \prod_{i=1}^{a} \hat{\mathrm{P}}(x_i \mid y). \tag{2}$$

We assume herein that NB and the other AnDE family members are implemented by compiling at training time a table of observed low-dimensional probabilities. Under this strategy, the complexity of building this model is $O(ta)$, where t is the number of training examples and a the number of attributes. As the model simply stores the frequency of each attribute value for each class after scanning the training examples, the space complexity is $O(kav)$, where k is the number of classes and v is the average number of attribute values. As the classifier only needs to estimate the probability of each class for the attribute values of the test case, the resulting complexity at classification time is $O(ka)$.

Despite the attribute independence assumption, NB delivers relatively accurate results. However, greater accuracy can be achieved if the attribute-independence assumption is relaxed. New algorithms based on NB have been developed, referred to as semi-Naive Bayesian techniques, that achieve greater accuracy by doing this, as real-world problems generally do have relationships among attributes [12].

Of numerous semi-naive Bayesian techniques, SP-TAN [7], Lazy Bayesian Rules (LBR) [13] and AODE [9] are among the most accurate. However, SP-TAN has very high computational complexity at training time and LBR has high computational complexity for classification. Contrastingly, AODE a more efficient algorithm, avoids some of the undesirable properties of those algorithms to achieve comparable results.

2.1 AODE

AODE extends NB's strategy of extrapolating from lower dimensional probabilities to make use of three-dimensional probabilities. It averages across over all of a class of three-dimensional models, which are called super-parent one-dependence estimators (SPODE). Each SPODE relaxes the attribute independence assumption of NB by making all other attributes independent given the class and one privileged attribute, the super-parent x_α.

AODE seeks to use

$$\hat{\mathrm{P}}(y, \mathbf{x}) = \sum_{\alpha=1}^{a} \hat{\mathrm{P}}(y, x_\alpha)\hat{\mathrm{P}}(\mathbf{x} \mid y, x_\alpha)/a. \tag{3}$$

In practice, AODE only uses estimates of probabilities for which relevant examples occur in the data. Hence,

$$\hat{\mathrm{P}}_{\mathrm{AODE}}(y, \mathbf{x}) = \begin{cases} \sum_{\alpha=1}^{a} \delta(x_\alpha)\hat{\mathrm{P}}(y, x_\alpha)\hat{\mathrm{P}}(\mathbf{x} \mid y, x_\alpha) / \sum_{\alpha=1}^{a} \delta(x_\alpha) & : \sum_{\alpha=1}^{a} \delta(x_\alpha) > 0 \\ \hat{\mathrm{P}}_{\mathrm{NB}}(y, \mathbf{x}) & : \text{otherwise} \end{cases} \tag{4}$$

where $\delta(x_\alpha)$ is 1 if attribute-value x_α is present in the data, otherwise 0. In other words, AODE averages over all super-parents whose value occurs in the data, and defaults to NB if there is no such parent.

2.2 A*n*DE

A*n*DE [10] generalises AODE's strategy of search free extrapolation from low-dimensional probabilities to high-dimensional probabilities. The first member of the A*n*DE family (where $n = 0$) is NB, the second member is AODE and the third is A2DE. Investigation into the accuracy of higher dimensional models with different training set sizes shows that a higher degree model might be susceptible to variance in a small training sample, and consequently that a lower degree model is likely to be more accurate for small data. On the other hand, higher degrees of A*n*DE may work better for larger training sets as minimizing bias will be of increasing importance as the size of the data increases [3].

For notational convenience we define

$$x_{i,j,\ldots,q} = \langle x_i, x_j, \ldots, x_q \rangle. \tag{5}$$

For example, $x_{2,3,4} = \langle x_2, x_3, x_4 \rangle$.

A*n*DE classifies using:

$$\hat{\mathrm{P}}_{\mathrm{A}n\mathrm{DE}}(y, \mathbf{x}) = \begin{cases} \sum_{s \in \binom{\mathcal{A}}{n}} \delta(x_s) \hat{\mathrm{P}}(y, x_s) \hat{\mathrm{P}}(\mathbf{x} \mid y, x_s) / \sum_{s \in \binom{\mathcal{A}}{n}} \delta(x_s) : \sum_{s \in \binom{\mathcal{A}}{n}} \delta(x_s) > 0 \\ \hat{\mathrm{P}}_{\mathrm{A}(n-1)\mathrm{DE}}(y, \mathbf{x}) \qquad : \text{otherwise.} \end{cases} \tag{6}$$

Attributes are assumed independent given the parents and the class. Hence, $\mathrm{P}(\mathbf{x} \mid y, x_s)$ is estimated by

$$\hat{\mathrm{P}}(\mathbf{x} \mid y, x_s) = \prod_{i=1}^{a} \hat{\mathrm{P}}(x_i \mid y, x_s) \tag{7}$$

Given sufficient training data, A2DE has lower error than AODE, but at the cost of significantly more computational resources.

3 Optimising Memory Consumption

In order to support incremental learning, A*n*DE classifiers compile a table of observed joint frequencies of attribute-value combinations during training. The frequencies table is used in testing to calculate posterior probabilities of class membership. The A*n*DE classifier requires the joint frequencies of n attribute value combinations per class. Additionally, as the classifier defaults to lower orders of n, for super-parents whose values do not occur in the data, the classifier also requires frequencies of all combinations of length up to n per class. As the space requirement for storing these joint frequencies for higher orders of n is undesirable, we developed a new representation that reduces the required space.

The frequency matrix for AODE is a 3-D matrix, where each cell holds the frequency of a (class, parent, child) combination. As an example, consider the frequency matrix for a dataset with two attributes (A and B). Attribute A has two values (a_1 and a_2), while attribute B has three values (b_1, b_2 and b_3).

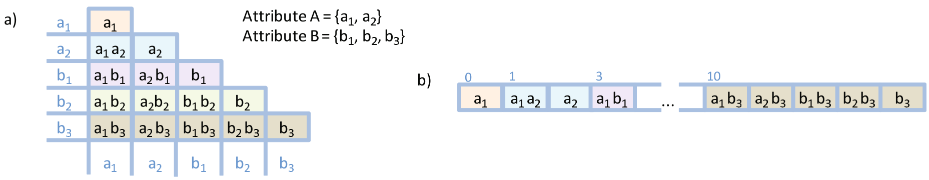


Fig. 1. AODE Parent Child Combinations

The parent and child dimensions of the frequency matrix is illustrated in Fig. 1a. It contains cells for each parent-child combination and the (n,n) locations are reserved for frequencies of parents. The 2-D structure is replicated for each class to form the 3-D frequency matrix for AODE.

The representation for A2DE is a 4-D matrix that is a collection of tetrahedral structures for each class. Each cell contains the frequencies of (class, parent1, parent2, child) combinations. The matrix reserves (class, parent1, parent1, parent1) cells for storing frequencies of class-parent1 combinations and (class, parent1, parent2, parent2) cells for storing class-parent1-parent2 combinations.

AnDE requires a matrix of $n + 2$ dimensions to store frequencies of all attribute value combinations. The outer dimension has k elements for each class. The n middle dimensions represent the n parent attribute values and the final dimension represents the child attribute values. The inner dimensions have av elements, where a is the number of attributes and v is the average number of attribute values (including missing values). Consequently, as the size of the frequency matrix is determined by figurate numbers ($P_{n+1}(av) = \binom{av+n}{n+1}$), resulting in a memory complexity of $O(k\binom{av+n}{n+1})$.

Although this representation allows for straight forward access of the frequency of a class-parent-child combination, the matrix has to be implemented as a collection of arrays. This incurs overhead and the does not guarantee that a contiguous block of memory is allocated for the matrix, reducing the possibility that required parts of the matrix are available in the system's cache.

The frequency matrix can be stored compactly with the elements of each row stored in consecutive positions. This representation minimises the overheads that can occur with multi-dimensional arrays. Taking AODE as an example, the rows in the 2-D matrix, which are all combinations involving the corresponding parent, can be stored sequentially in a 1-D array as shown in Fig. 1b.

Allocating slots for all combinations of attribute values in the frequency matrix simplifies access. However, this produces a sparse matrix containing unused slots allocated for impossible combinations. As training and testing cases have only single valued attributes, combinations of attribute values of the same attribute are impossible. In the case of the AODE example, the frequency matrix contains slots to record frequencies of a_1a_2, b_1b_2, b_1b_3 and b_2b_3, which are impossible combinations (shaded in black in Fig. 2a). The size of the frequency matrix can be reduced by avoiding the allocation of memory for such impossible combinations. In the AODE example, the size of the 2-combinations matrix can be reduced from 10 to 6. The size of the n combinations matrix is $\binom{a}{n+1}v^{n+1}$.

To avoid allocating space for impossible combinations and simplify indexing, the frequency matrix is decomposed into a series of structures for storing attribute value combinations of a specific length. Taking the AODE example, the set of 1-D arrays for storing only possible attribute value combinations is shown in Fig. 2b. Array `freq1` contains frequencies of each attribute value and Array `freq2` contains frequencies of all valid attribute value pairs.

4 Optimising Testing Time

AnDE classifies a test instance by calculating posterior probabilities of class membership. They are calculated by iterating through all parent-child permutations, resulting in a time complexity of $O(ka^{(n+1)})$. We reduced the overall testing time by reorganising the frequency matrix and the looping structure to taking advantage of locality of reference.

The CPU cache is a fast but limited memory resource, which stores copies of most frequently used data. It is used to reduce average time to access memory. We reorganized the memory representation and minimized data retrieval from memory to improve the likelihood of availability of data in the CPU cache.

The compact memory representation for the frequency matrix is a 2-D array, which contains k copies of arrays that record n-combination attribute value frequencies per class. For example, Fig. 3a illustrates the memory representation for a dataset with two attributes (A and B) and two classes (c_1 and c_2). The 2-D representation is poorly suited for accessing all class frequencies of some attribute-value combination. Especially, in the case of datasets with large collection of attributes, this representation reduces the likelihood of all the per-class frequencies of some attribute value combination being available in the system's high-speed access cache.

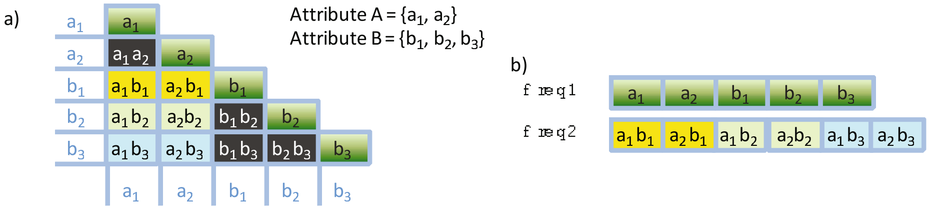


Fig. 2. Valid AODE Parent Child Combinations

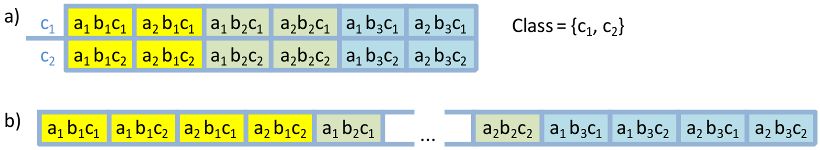


Fig. 3. Storing per Class Frequencies in Sequence

The locality of reference of attribute-value combinations for all classes can be improved by storing them next to each other. Taking AODE as an example, the 2-D frequency matrix (Fig. 3a) can be represented in a 1-D array by interleaving the per class frequencies as shown in Fig. 3b. This representation improves the chances of the frequencies of attribute-value combinations for both the classes being available in high-performance memory. In order to take full advantage of locality of reference of class frequency combinations the looping structure of the classifier also had to be rearranged from looping through each class, parent and child to loop through each parent, child and class.

AnDE requires conditional probabilities for all parent child permutations. Iterating through all permutations requires all relevant offsets to be retrieved, indexes to be calculated and the relevant frequencies to be retrieved. Although these retrievals would be loaded into the CPU cache, they are only used once. In order to reuse data and improve the likelihood of data being available in the CPU cache, we modified the implementation to only iterate over unique combinations. During each iteration conditional probabilities for all permutations of each combination are calculated. This results in reducing the iterations from $ka^{(n+1)}$ to $ka^{(n)}$ and reducing the total number of memory accesses.

The conditional probability of a parent-child attribute permutation is calculated by dividing the frequency of parent-child attributes occurring together by the frequency of parents. The numerator is constant for all permutations of a parent combination. The improved implementation also allows this numerator to be reused, reducing the amount of frequency fetches and the number of index calculations. Overall, this reduces the number of frequency accesses of parent-child attribute value combinations to $\frac{1}{2}$ for AODE, $\frac{1}{3}$ for A2DE and $\frac{1}{4}$ for A3DE.

5 Evaluation

The effectiveness of the improvements to reduce memory usage and testing times were evaluated on a collection of Datasets from the UCI machine learning repository[1]. The evaluation was focused on three members of the AnDE family of algorithms: AODE, A2DE and A3DE. Although NB is the first member of the AnDE family, it was not evaluated as the improvements are unlikely to have any impact. The improvements were compared against the Weka version of AODE and naive versions of A2DE and A3DE.

5.1 Test Environment

We selected nine datasets, described in Table 1, from the UCI machine learning repository for the comparisons. The chosen collection includes small, medium and large datasets with small, medium and high dimensionality. The datasets were split into two sets, with 90% of the data used for training and the remaining 10% used for testing. The experiments were conducted on a single CPU single core virtual Linux machine running on a Dell PowerEdge 1950 with dual quad core Intel Xeon E5410 processor running at 2.33GHz with 8 GB of RAM.

Table 1. Datasets used for experiments

Dataset	Cases	Att	Values	Classes
Abalone	4177	8	24	3
Adult	48842	14	117	2
Connect-4	67557	42	126	3
Covertype	581012	54	118	7
Dermatology	366	34	132	6
House Votes 84	435	16	48	2
Sonar	208	60	180	2
SPAM E-mail	4601	57	171	2
Waveform-5000	5000	40	120	3

The implementations of the three algorithms of the AnDE family are limited to categorical data. Consequently, all numerical attributes are discretized. When MDL discretization [5], a common discretization method for NB, was used within each cross-validation fold, we identified that many attributes have only one value. So, we discretized numerical attributes using three-bin equal-frequency discretization prior to classification for these experiments.

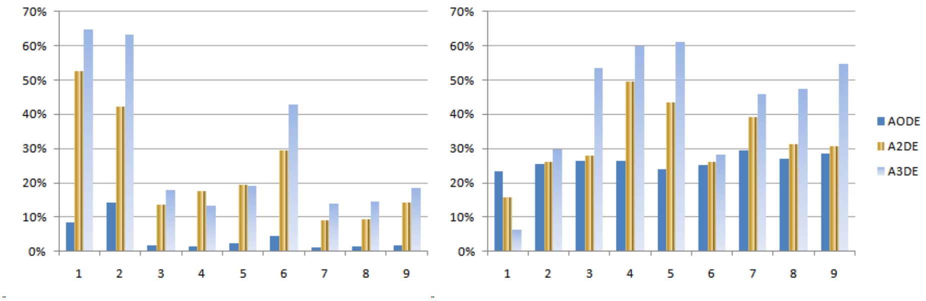
The memory usage of the classifier was measured by the 'Classmexer' tool [4], which uses Java's instrumentation framework to query the Java virtual machine (JVM). It follows relations of objects, so that the size of the arrays inside arrays are measured, including their object overhead and padding.

Accurately measuring execution time for the Java platform is difficult. There can be interferences due to a number of JVM processes such as garbage collection and code profiling. Consequently, to make accurate execution time measurements, we use a Java benchmarking framework [2] that aims to minimize the noise during measurements. The framework executes the code for a fixed time period (more than 10 seconds) to allow the JVM to complete all dynamic optimizations and forces the JVM to perform garbage collection before measurements. All tests are repeated in cases where the code is modified by the JVM. The code is also executed a number of times with the goal of ensuring the cumulative execution time to be large enough for small errors to be insignificant.

5.2 Optimised Memory Consumption

The memory usage for AnDE was reduced by the introduction of a new data structure that avoids the allocation of space for impossible combinations. The reductions in memory usage for the enhanced AnDE implementations were compared against the respective versions of AnDE that stores the frequency matrix in a single array. We do not present the memory reductions of compacting the multi-dimensional array into one dimension as they are specific to Java.

The reductions in memory usages are summarised in Fig. 4a. Results show that the memory reduction for AODE ranged from 1% to 14%. The highest percentage in reduction was observed for the adult dataset, which had a reduction of 9.67KB. The main reason for the large reduction is the high average number of attribute values of 8.36 for the adult dataset. In contrast, the other datasets have average number of attribute values of around 3.



(a) Memory Usage (b) Mean Testing Times

Fig. 4. Proportions of Reductions in Memory Usage and Testing Times

The memory usage for A2DE was reduced by a minimum of 9% to a maximum of 53%. The maximum amount of reduction in memory was observed for the high-dimensional Covertype dataset, which had a reduction of around 4.68MB.

The enhanced version of A3DE resulted in the highest reduction in memory usage with reductions ranging from 13% to 64%. The reductions in memory usage for the high dimensional Covertype, Dermatology, Sonar Classification and SPAM E-mail datasets were over 100MB.

5.3 Optimised Testing

The total testing times of the algorithms were compared using the test environment. The proportions of reduction in mean test times for AnDE are given in Fig. 4b. The optimisations result in reductions in average testing times for all three algorithms. The reductions for A3DE were highest, with a 61% (0.83s) mean reduction for the small but high-dimensional Dermatology dataset and 60% (8.89ks) reduction for the large and high-dimensional Covertype dataset. The improvements also reduced the testing times of low dimensional datasets of Abalone (6%) and House Votes (28%).

The reductions in testing times were substantial for A2DE, with reductions ranging from 16% (for Abalone) to 50% (Covertype). The improvements to AODE also resulted in reduced total execution times ranging from 23% to 30%. The highest percentage of reduction was exhibited for the dataset with the largest number of attributes, Sonar Classification.

6 The Evaluation of A3DE

We evaluated the classification accuracy of AnDE algorithms comparing how their performance varies as n increases within the AnDE framework. Previous research [10] has compared the effectiveness of AODE to A2DE, but only limited experimental results were presented for A3DE as the Weka implementation failed on high dimensional datasets due to its high memory requirements.

We compared the effectiveness the AnDE members using the enhanced versions implemented in the Weka workbench on the 62 datasets that were used to evaluate the performance of A2DE [10]. Each algorithm was tested on each dataset using the repeated cross-validation bias-variance estimation method [8]. We used two-fold cross validation to maximise variation in the training data between trials. In order to minimise the variance in our measurements, we report the mean values over 50 cross-validation trials.

The experiments were conducted on the same virtual machine used to evaluate the effectiveness of the improvements. Due to technical issues, including memory leaks in the Weka implementation, increasing amounts of memory is required when multiple trials are conducted. Consequently, we were unable to get bias-variance results for four datasets (Audiology, Census-Income, Covertype and Cylinder bands), that were of high dimensionality. We compared the relative performances of AODE, A2DE and A3DE on the remaining 58 datasets. The lower, equal or higher outcomes when the algorithms are compared to each other is summarised as win/draw/loss records in Tab. 2.

The results show that the bias decreases as n increases at the expense of increased variance. The bias of A3DE is lower significantly more often than not in comparison to A2DE and AODE. The bias of A2DE is lower significantly more often relative to AODE. In contrast, the variance of AODE is lower significantly more often than A2DE and A3DE. The variance of A2DE is lower significantly more often relative to A3DE.

None of the three algorithms have a significantly lower zero-one loss or RMSE on the evaluated datasets. We believe that this is due to the wide range sizes of datasets used in the evaluation. We hypothesize that members of the AnDE family with lower n, that have a low variance, are best suited for small datasets. In contrast, members with higher degrees of n are best suited for larger datasets.

6.1 A3DE Performance on Large Datasets

To assess the hypothesis that increasing values of n within the AnDE family are suited to increasing data quantity, we compared A3DE to lower-order family members on datasets with over 10,000 cases. Out of the 58 datasets, seven datasets (Adult, Connect-4 Opening, Letter Recognition, MAGIC Gamma Telescope, Nursery, Pen Digits and Sign) satisfied this criterion. The number of cases

Table 2. Win/Draw/Loss: AnDE, n = 1, 2 and 3 on 58 data sets

	A3DE vs A2DE		A3DE vs AODE		A2DE vs AODE	
	W/D/L	p	W/D/L	p	W/D/L	p
Bias	34/3/21	**0.052**	40/1/17	**0.002**	43/0/15	<**0.001**
Variance	17/2/39	**0.002**	15/2/41	<**0.001**	16/1/41	<**0.001**
Zero-one loss	24/3/31	0.209	24/2/32	0.175	29/2/27	0.447
RMSE	24/3/31	0.209	28/0/30	0.445	31/1/26	0.298

Table 3. Win/Draw/Loss: AnDE, n=1,2 and 3 on large data sets

	A3DE vs A2DE		A3DE vs AODE	
	W/D/L	p	W/D/L	p
Bias	6/0/1	0.063	7/0/0	**0.008**
Variance	2/0/5	0.227	1/0/6	0.063
Zero-one loss	7/0/0	**0.008**	7/0/0	**0.008**
RMSE	7/0/0	**0.008**	7/0/0	**0.008**

of the chosen datasets ranged from just over 10,000 cases (Pen Digits) to over 60,000 cases (Connect-4 Opening).

The evaluation results are summarised as win/draw/loss records in Table 3. As expected, the results show A3DE has a lower bias and higher variance than A2DE and AODE. The zero-one loss and the RMSE of A3DE are lower for all the evaluated datasets in comparison to A2DE and AODE (p=0.008). These results confirm that A3DE performs better than its lower-dimensional variants at classifying larger datasets.

7 Conclusions

The AnDE family of algorithms perform search-free learning. The parameter n controls the bias-variance trade-off such that $n = a$ provides a classifier whose asymptotic error is the Bayes optimum. We presented techniques for reducing the memory usage and the testing times of the AnDE implementations that make A3DE feasible to employ for higher-dimensional data. As A3DE is superior to AnDE with lower values of n when applied to large data, and as the linear complexity and single pass learning of AnDE make it particularly attractive for learning from large data, we believe these optimizations have potential for considerable impact.

We developed a new compact memory representation for storing the frequencies of attribute-value combinations that stores all frequencies in a 1-D array avoiding the allocation of space for impossible attribute-value combinations. The evaluation results showed that the enhancements substantially reduced the memory requirements. The enhancements reduced the overall A3DE memory requirements ranging from 13% to 64%, including reductions of over 100MB for the high-dimensional datasets.

The classification times of the AnDE algorithms were improved by reorganising the memory representation to maximise locality of reference and minimising memory accesses. These enhancements resulted in substantial reductions to the total testing times for the AnDE family of algorithms. In the case of A3DE, the maximum reduction in total testing time was 8.89ks, which was a reduction of 60%, for the Covertype dataset.

The enhancements to the AnDE algorithms opened the door for evaluating the performance of A3DE. As expected, the results showed that A3DE has lower bias in comparison to A2DE and AODE. The results for zero-one error between A3DE, A2DE and AODE did not produce a clear winner. However, A3DE produced the lowest error for large datasets (with over 10,000 cases).

The computational complexity of AnDE algorithms is linear with respect to the number of training examples. Their memory requirements are dictated by the number of attribute values in the data. Consequently, the most accurate and feasible member of the AnDE algorithm for a particular dataset will have to be decided based on the dataset's size and its dimensionality.

References

1. Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases, `http://www.ics.uci.edu/~mlearn/MLRepository.html`
2. Boyer, B.: Robust Java benchmarking (2008), `http://www.ibm.com/developerworks/java/library/j-benchmark1.html`
3. Brain, D., Webb, G.I.: The Need for Low Bias Algorithms in Classification Learning From Large Data Sets. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) PKDD 2002. LNCS (LNAI), vol. 2431, pp. 62–73. Springer, Heidelberg (2002)
4. Coffey, N.: Classmexer agent, `http://www.javamex.com/classmexer/`
5. Fayyad, U., Irani, K.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proc. of the 13th Int. Joint Conference on Artificial Intelligence, pp. 1022–1029. Morgan Kaufmann (1993)
6. Hui, B., Yang, Y., Webb, G.I.: Anytime classification for a pool of instances. Machine Learning 77(1), 61–102 (2009)
7. Keogh, E., Pazzani, M.: Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. In: Proc. of the International Workshop on Artificial Intelligence and Statistics, pp. 225–230 (1999)
8. Webb, G.I.: Multiboosting: A technique for combining boosting and wagging. Machine Learning 40(2), 159–196 (2000)
9. Webb, G.I., Boughton, J., Wang, Z.: Not so naive Bayes: Aggregating one-dependence estimators. Machine Learning 58(1), 5–24 (2005)
10. Webb, G.I., Boughton, J., Zheng, F., Ting, K.M., Salem, H.: Learning by extrapolation from marginal to full-multivariate probability distributions: Decreasingly naive Bayesian classification. Machine Learning 86(2), 233–272 (2012), doi:10.1007/s10994-011-5263-6
11. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2005)
12. Zheng, F., Webb, G.I.: A comparative study of semi-naive Bayes methods in classification learning. In: Simoff, S.J., Williams, G.J., Galloway, J., Kolyshakina, I. (eds.) Proc. of the 4th Australasian Data Mining Conference (AusDM 2005), pp. 141–156 (2005)
13. Zheng, Z., Webb, G.I.: Lazy learning of Bayesian rules. Machine Learning 41(1), 53–84 (2000)

An Aggressive Margin-Based Algorithm for Incremental Learning

JuiHsi Fu[⋆] and SingLing Lee

National Chung Cheng University
168 University Road, Minhsiung Township,
Chiayi 62162, Taiwan, R.O.C.
{fjh95p,singling}@cs.ccu.edu.tw

Abstract. In incremental learning, the classification model is incrementally updated using the small datasets. Different with existing methods, our approach updates the current classifier according to each sample in the dataset, respectively. The classifier is updated by adjusting more than the margin of each sample. Then the new classifier is generated by carefully analyzing classifier adjustments caused for labeled samples. Additionally the new classifier shall correct prediction mistakes of the previous classifier as many as possible. In details, we formulate simple constrained optimization problems and then the updated classifier is the solution derived using Lagrange multipliers. In our experiments, 13 real-world dataset are used to present the effectiveness of the proposed approach. The experimental results are shown that our update strategy is able to adjust the classifier properly. And it is also shown that the proposed incremental learning approach is suitable to be applied for the requirement of frequently adjusting the existing classifiers.

Keywords: Incremental Learning, Margin-based Approaches, Passive-Aggressive (PA) Algorithm, Period Datasets, Classifier Adjustment.

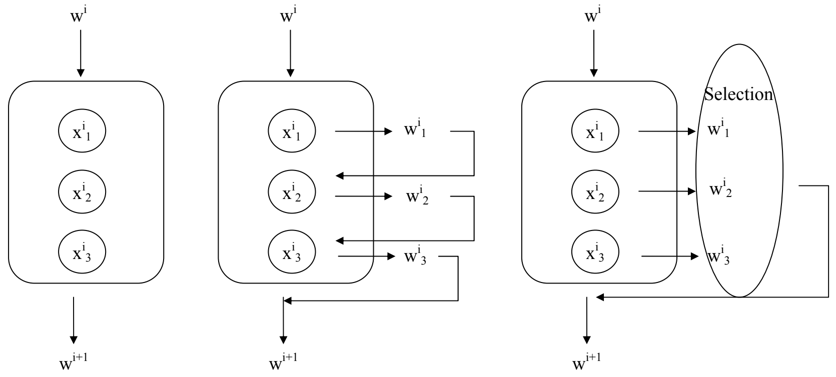
1 Introduction

Requests of analyzing collected period data have been emerged in recent practical applications that includes network traffic analysis [1], anomaly detection [2], and intrusion detection [3]. Generally, those applications are implemented for adjusting classifiers/detectors periodically. Most of incremental learning approaches have been proposed based on decision-tree [4], neural network [5,6], and Support Vector Machines (SVM) [3,7,8,9,10]. Typically they are designed to build the statistic classification model based on the previously seen samples and to correct its prediction mistakes on new labeled samples. While focusing on the sample space, SVM generalizes the separating hyperplane (classifier) based on the whole sample distribution, and maximizes the margins of labeled samples (support vectors). The margin of a sample is a distance between the separating hyperplane. And SVM is theoretically proven that

[⋆] This work is supported by NSC, Taiwan, under grant no. NSC 99-2218-E-492-006.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 62–73, 2012.

the hyperplane is able to well separate samples with different labels. In [10], an incremental batch SVM approach was designed to update the classifier by solving a constrained optimization problem based on each set of collected samples. An example is illustrated in Fig. 1 (a) where the classifier w^i is adjusted as w^{i+1} depending on the set of samples, $\{x_1^i, x_2^i, x_3^i\}$. This approach should solve a complicated constrained optimization problem since those collected samples are adopted simultaneously. Other approaches [8,9] adjusted SVM classifiers incrementally by identifying each new sample as a support vector or not. Different with [10], in Fig. 1 (b) the classifier w^i is adjusted as w_1^i using first sample x_1^i in the set, and then w_1^i is updated as w_2^i using x_2^i. Thus w^i is incrementally adjusted as w^{i+1} depending on each sample in the set. The advantage of [8,9] is to maintain useful samples that were previously seen as support vectors and to obtain efficient update steps without solving a constrained optimization problem. But in those SVM approaches, the hyperplanes might not be quickly adjusted when encountering diverse sample distribution. In other words, the diverse samples have small chances to be support vectors because the distribution of those samples is significantly different with the distribution of samples in the set. Thus in this paper, our approach is to simplify the constrained optimization problem for update steps and to adapt the diverse sample distribution for classifiers.



(a) w^i is adjusted by all samples simultaneously.

(b) w^i is adjusted incrementally by each sample.

(c) w^i is adjusted by one sample in the set.

Fig. 1. Concepts of solving problems of adjusting classifiers. w^i and w^{i+1} are the current classifier and the next one. x_1^i, x_2^i, and x_3^i are samples used for adjusting w^i.

Rather than training the SVM classifier based on each sample or each set of collected samples, our approach adjusts the current classifier incrementally according one sample in each collected set. Thus for each potential update, we formulate an optimization problem with single constraint. Additionally our updated classifier shall correct prediction mistakes of the previous classifier as many as possible. Compared with [10], we divide a complicated constrained optimization problem into several simpler ones. In other words, the classifier is adjusted as several potential ones depending on different samples. An example is illustrated in Fig. 1 (c). The classifier w^i is adjusted as w_1^i, w_2^i, and w_3^i respectively using x_1^i, x_2^i, and x_3^i. And then the classifier that adjusts the most w^i's mistakes is selected as the next w^{i+1}. In this paper, we are motivated by the simplicity of online Passive-Aggressive (PA) algorithm [11]. One sample's margin is selected as the basis for classifier adjustment. Thus in our approach, while a sample is used for updating and its sign is incorrectly predicted, the classifier adjustment is aggressively achieved within the margin. Additionally the updated classifier shall correct prediction mistakes of the previous classifier as many as possible. In this paper, we formulate a simple constrained optimization problem for each sample and then the candidate updated classifier is the solution derived using Lagrange multipliers. It is noted that, we get a closed form solution for each potential updated classifier. Particularly the selected new classifier, updated by the suitable margin, shall obtain the best classification accuracy on the collected dataset. It is expected that, this selection strategy is able to avoid the new classifier being extremely specific to the previous one. And the updated classifier could flexibly adapt the diverse sample distribution because there is no need for the proposed approach to maintain previously seen samples.

Basically PA has the ability to frequently update the classifiers, but its two straightforward approaches may not be able to achieve impressive results. Firstly, PA update steps are specific to each labeled sample whether it is inconsistent or not. The consequence is that updated classifiers would obtain the unstable prediction ability. Secondly, the other PA approach is to update the classifier respectively using each sample. Then the selected classifier among all updated ones shall have the best classification accuracy on the collected dataset. Compared with our proposed approach, this approach does not actively correct prediction mistakes of the previous classifier. Thus these two approaches do not fully utilize the learning knowledge in each collected dataset. Moreover our approach is similar with re-sampling approaches, like bagging [12], to obtain improved classification accuracy by depending on subsets of the sample set. The major difference is that, we focus on designing efficient update steps for online applications so that a closed form solution for the updated classifier could be obtained.

The rest of our paper is organized as follows. The online PA algorithm is reviewed in Section 2. In Section 3, we detailedly describe the proposed approach and build the mathematic model. Experimental results are presented in Section 4. Finally, we conclude the paper in Section 5.

2 Online Passive-Aggressive Algorithm

In online learning, each training sample is discarded after it is used to update the classifiers. Some research works like the Perceptron algorithm [13,14,15] and margin-based approaches [16,17] have been proven to be effective in a board range of applications. Additionally it is worth noting the Passive-Aggressive

(PA) Algorithm [11] is a margin-based online learning approach that could be applied for various prediction tasks. PA uses linear predictors for label prediction of each incoming sample. And each update step of PA is executed depending on the margin of the labeled sample. The objective of PA update is to adjust the previous classifier as less as possible while the condition of classifier adjustment is satisfied. At the round t, let w^t be the vector of weights, x^t be the sample, $y^t \in \{+1, -1\}$ be x^t's true label, and the term $y^t(w^t \cdot x^t)$ be the signed margin. The new classifier w^{t+1} is the solution to the following constrained optimization problem,

$$w^{t+1} = argmin_{w \in R^n} \frac{1}{2}||w - w^t||^2 \quad s.t. \quad l(w, (x^t, y^t)) = 0, \tag{1}$$

where $l(w, (x^t, y^t))$ is the hinge loss of w's prediction on x^t.

$$l(w, (x, y)) = \begin{cases} 0, & y(w \cdot x) \geq 1 \\ 1 - y(w \cdot x), & \text{otherwise} \end{cases} \tag{2}$$

Typically whenever the loss is zero, PA is *passive* and $w^{t+1} = w^t$ means no classifier adjustment. And while the loss is positive (less than 1), w^t is *aggressively* updated by adjusting more than the margin, $y^t(w^t \cdot x^t)$, and then the constrain $l(w^{t+1}, (x^t, y^t)) = 0$ can be satisfied. Then the Lagrangian of the optimization problem in Eq. (1) is defined as Eq. (3).

$$L(w, \tau) = \frac{1}{2}||w - w^t||^2 + \tau(1 - y^t(w \cdot x^t)) \tag{3}$$

Let the partial derivation of l with respect to w be zero and then let the deviation of τ with respect to τ be zero, we have

$$w = w^t + \tau y^t x^t$$
$$\tau = \frac{1 - y^t(w^t \cdot x^t)}{||x^t||^2}$$

Ultimately the PA update is performed by solving the constrained optimization problem in Eq. (1). And it is theoretically shown that the aggressive update strategy of PA modifies the weight vector as less as possible. The effectiveness of PA in solving problems of classification and regression is formally analyzed in [11]. Based on this well-defined learning model of PA, several online algorithms [18,19] have been proposed for adding confidence information and handling non-separable data.

3 Incremental Passive-Aggressive Learning Algorithm

While each set of labeled period samples comes, the existing classifier shall be periodically updated for adapting the latest sample distribution. In this paper, we propose an incremental learning algorithm, named Incremental Passive-Aggressive (IPA). It adjusts the current classifier incrementally using one sample in each collected set. For each potential sample, there are two update steps in IPA: 1) to correct prediction mistakes of the current classifier, and 2) to aggressively update the current classifier by adjusting more than the margin. At last, the error minimization classifier on the collected dataset is selected as the next classifier. Before formulating the model of the proposed approach, we define some notations. Given the labeled dataset K^t collected at the round t, there are $|K^t|$ sample-label pairs, $\{(x_1, y_1), ..., (x_{|K^t|}, y_{|K^t|})\}$. w^t is the classifier at the round t, the vector of weights. When using each labeled sample $x_k \in K^t$, the updated classifier w^{t+1} shall correct mistakes of the previous classifier w^t as many as possible and w^t shall be adjusted as less as possible. Aggressively, if x_k obtains the incorrect predicted sign from w^t, then the adjustment for w^t should be achieved within more than x_k's margin. Thus these update steps to w^t are formulated as the constrained optimization problem,

$$\begin{aligned} f(w^t, (x_k, y_k), K^t) \;&=\; argmin_{\overline{w} \in R^n}\{\frac{1}{2}||\overline{w} - w^t||^2 \\ &+\; C_0 \sum_{x_i \in K^t, x_i \neq x_k} l(\overline{w}, (x_i, y_i))\} \\ & s.t.\ l(\overline{w}, (x_k, y_k)) = 0, \end{aligned} \tag{4}$$

where C_0 is a constant to control the tradeoff between the classifier deviation and the corrected prediction mistakes, and $l(\overline{w}, (x_i, y_i))$ is the hinge loss function.

Furthermore, after w^t is updated using every sample $x_k \in K^t$ according to Eq. (4), those updated classifiers, $\{f(w^t, (x_k, y_k), K^t) : 1 \leq k \leq |K^t|\}$, are the candidates for the new classifier. In order to avoid the new classifier being extremely specific to the current classifier, the selection strategy is to find the proper classifier which has the most accurate classification performance on K^t. When more than one updated classifiers have the highest classification accuracy, we select the updated classifier which has the smallest difference with w^t. Hence the new classifier w^{t+1}, selected among the candidate set of the updated classifiers, is the solution to the optimization problem,

$$w^{t+1} = argmin_{w \in \{f(w^t, (x_k, y_k), K^t)\ :\ 1 \leq k \leq |K^t|\}} C \sum_{x_i \in K^t} l(w, (x_i, y_i)) + ||w - w_t||, \tag{5}$$

where C is a large constant in order to select w strongly depending on the errors.

To solve the problem in Eq. (4), let $C_0 = 1$ and κ^t, the subset of $|K^t|$, be the set of samples of which predicted labels are incorrectly decided by w^t. While the loss of each sample in κ^t is positive (less than 1), the Lagrangian of the constrained optimization problem is defined as Eq. (6):

$$L(\overline{w}, \tau) = \frac{1}{2}||\overline{w} - w^t||^2 + \sum_{x_i \in \kappa^t, x_i \neq x_k} (1 - y_i(\overline{w} \cdot x_i)) + \tau(1 - y_k(\overline{w} \cdot x_k)) \tag{6}$$

Let the partial derivation of l with respect to $\overline{w}$ be zero,

$$\begin{aligned}\nabla_{\overline{w}} L(\overline{w},\tau) &= \overline{w} - w^t - \sum_{x_i \in \kappa^t, x_i \neq x_k} y_i x_i - \tau y_k x_k \\ => \overline{w} &= w^t + \sum_{x_i \in \kappa^t, x_i \neq x_k} y_i x_i + \tau y_k x_k \end{aligned} \tag{7}$$

Then substituting Eq. (7) into Eq. (6), we have

$$\begin{aligned} L(\tau) = \frac{1}{2} || & \sum_{x_i \in \kappa^t} y_i x_i + \tau y_k x_k ||^2 \\ & + \sum_{x_i \in \kappa^t, x_i \neq x_k} (1 - y_i((w^t + \sum_{x_i \in \kappa^t, x_i \neq x_k} y_i x_i + \tau y_k x_k) \cdot x_i)) \\ & + \tau (1 - y_k((w^t + \sum_{x_i \in \kappa^t, x_i \neq x_k} y_i x_i + \tau y_k x_k) \cdot x_k)) \end{aligned} \tag{8}$$

At last let the deviation of Eq. (8) with respect to τ be zero,

$$\begin{aligned} 0 &= -\tau y_k^2 ||x_k||^2 + (1 - y_k(w^t \cdot x_k)) - y_k x_k \sum_{x_i \in \kappa^t, x_i \neq x_k} y_i x_i \\ => \tau &= \frac{1 - y_k(w^t \cdot x_k) - y_k x_k \sum_{x_i \in \kappa^t, x_i \neq x_k} y_i x_i}{||x_k||^2} \end{aligned} \tag{9}$$

Ultimately, each update of the proposed incremental learning algorithm is performed by solving the constrained optimization in Eq. (4) and the updated classifier is determined by solving Eq. (5). It is theoretically presented in Eq. (7) and (9) that the update to the current classifier w^t is performed by correcting its prediction mistakes κ^t, and by adjusting it within the margin when the sample is incorrectly predicted. Overall the proposed algorithm is presented in Algorithm 1. At each round t, the dataset K^t is collected to update the current classifier w^t. And the samples of which predicted labels are incorrectly assigned by w^t are identified as κ^t, at line 4-5. Then for each sample $x_k \in K^t$, the current classifier w^t is individually updated as the candidate classifier $\overline{w_k}$ according to Eq. (7) and (9), at line 7-8. At last, the classifier $\overline{w_k}$ is selected as w^{t+1} if it gains the least prediction errors on K^t, at line 10. Particularly at the first round, w^1 is initialized as $(0, ..., 0)$ and its prediction result is always positive. Thus the w^1 is adjusted as the first updated classifier w^2 depending on the false positive sample that could cause the minimum $||w^2 - w^1||$. Moreover in addition to minimizing the classifier deviation, we correct mistakes of the previous classifier. In terms of convergence, each classifier is adjusted as small as possible. Also it is expected that, our approach is able to adaptively enhance the degree of adjusting classifiers when encountering diverse sample distribution that would cause significant prediction losses.

Algorithm 1. Incremental PA Learning Algorithm

```
input : $C_0$
Initialize: $w^1 = (0, ..., 0)$, $C = 10,000$ ;
for $t$=1,2,... do
    receive the collected labeled dataset $K^t$ ;
    predict $\widehat{y_x}$=sign($w^t \cdot x_k$) for each $x_k \in K^t$ ;
    collect $\kappa^t = \{x_k | x_k \in K^t \textit{ and } y_x \neq \widehat{y_x}\}$ ;
    for each $x_k \in K^t$ do
        set $\tau_k = \frac{1 - y_k(w^t \cdot x_k) - y_k x_k \sum_{x_i \in \kappa^t, x_i \neq x_k} y_i x_i}{||x_k||^2}$ ;
        update $\overline{w_k} = w^t + \sum_{x_i \in \kappa^t, x_i \neq x_k} y_i x_i + \tau y_k x_k$ ;
    end
    select $w^{t+1} = argmin_{w \in \{\overline{w_k} \,:\, 1 \leq k \leq |K^t|\}} C \sum_{x_i \in K^t} l(w, (x_i, y_i)) + ||w - w^t||$ ;
end
```

4 Experiments

In this section, our experiments are designed to present the performance of our approach in classification accuracy while the classifier is incrementally updated by several small training sets. To present the effectiveness of updating classifiers in our approach, we also implement the online PA and an incremental batch SVM [9]. Additionally in order to show the effectiveness of correcting mistakes of the previous classifier in eq. (4), the performance of our approach with $C_0 = 0$ is also compared in following experiments. In terms of evaluating classification accuracy of a classifier, we would like to significantly present classification results of samples in two different classes. We use the measurement of micro-average accuracy to average the classification accuracies that are calculated in two classes, respectively. For consistence, the summations of loss errors in the eq. (4) and (5) are also revised as (1 - micro-average accuracy).

Table 1 presents 13 real-world data collections from 4 different sources used in our experiments. The *multi-domain sentiment dataset* [1] contains product reviews downloaded from Amazon.com from four product types (domains): Kitchen, Books, DVDs, and Electronics. Each domain has several thousand reviews, but the exact number varies by domain. In this experiment, only Books, DVDs are used for evaluating performance of those learning approaches. From the second data source, the dataset at *ECML/PKDD-2006 discovery challenge*[2] is used to decide whether received emails are spam or non-spam. Especially there are over 10,000 features in those three datasets, Books, DVDs, and Emails. But it is difficult to analyze performance of the SVM classifiers implemented in Matlab [9] because the execution is time consuming on those high dimensional datasets. Thus we randomly select a part of documents, as presented in Tab. 1, in following experiments. From the third data source, *Spamming Bots* [20] is the set of response codes of the sent emails, collected in National Chung Cheng

[1] Sentiment. http://www.cs.jhu.edu/ mdredze/datasets/sentiment/

[2] ECML/PKDD-2006. http://www.ecmlpkdd2006.org/challenge.html

Table 1. 10 real-world datasets: sizes of the classes and the size of feature dimensions

Dataset	Source	Class(size)	Class(size)	Dimensions
DVDs	Sentiment	positive(292)	negative(300)	1488
Books	Sentiment	positive(289)	negative(287)	1548
Emails	ECML/PKDD	spam(210)	non-spam(445)	1034
Connectionist Bench	UCI	1(111)	2(97)	60
Ionosphere	UCI	b(126)	g(225)	34
German	UCI	Good(700)	Bad(300)	23
Australian Credit Approval	UCI	0(383)	1(307)	14
Statlog (Heart)	UCI	1(150)	2(120)	13
yeast	UCI	CYT(463)	ME1(44)	9
abalone	UCI	10(634)	4(57)	8
Pima Indians Diabetes	UCI	0(500)	1(268)	8
ecoli	UCI	cp(143)	im(77)	8
Spamming Bots	CCU	normal(1560)	spamming(150)	5

University (CCU). It is used to analyze the behavior of each email sender and then to detect the spamming bots. At last the other datasets are the benchmarks in the *UCI repository* [3]. While we evaluate classification performance of learning approaches, we randomly divide each dataset into 10 subsets, and one of subsets is received at each round. In other words, one subset is used for initially training the classifier and deciding the value of C_0 in eq. (4) by obtaining the highest classification accuracy on the first subset. Then others are received at each of 9 rounds. The classification accuracy at each round is measured by classification results of the classifier updated at previous rounds. To reduce variability in experimental results, we arrange 10 subset-round permutations on each dataset and average those 10 classification accuracies at each round.

At first these experiments, except on *Diabetes* in Fig. 2, are demonstrated that the proposed IPA has better performance than IPA with $C_0 = 0$. That means, in addition to minimizing the classifier deviation, it is effective in eq. (4) to correct mistakes for updating the previous classifier. And on *Diabetes*, correction of mistakes to the classifier could not improve the classification accuracy on latter samples. It seems, on *Diabetes* previous learning knowledge is not useful for latter label prediction. Secondly on *Australian*, *Ionosphere*, *Bots*, and *10+4* in Fig. 3-4, it is presented that the online PA method can not obtain the remarkable classification performance since its update strategy is specific to each labeled sample. That means, the online PA method tends to be updated by inconsistent samples. Furthermore, except experimental results on *Australian* and *Ionosphere* in Fig. 3, it is shown that our approach obtains the best (or similar) classification accuracy in comparison with other approaches. We update the classifier by carefully analyzing classifier adjustment caused for the labeled dataset. Then the remarkable classification accuracy is obtained at each round after the classifier is incrementally updated on most of datasets. Also it is shown

[3] UCI Repository. http://archive.ics.uci.edu/ml/

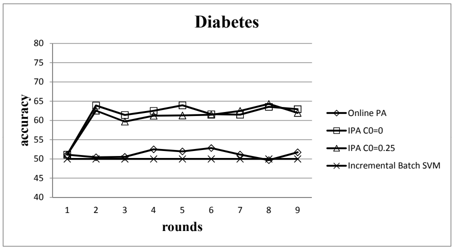


Fig. 2. Classification results of incremental learning approaches on *Diabetes*

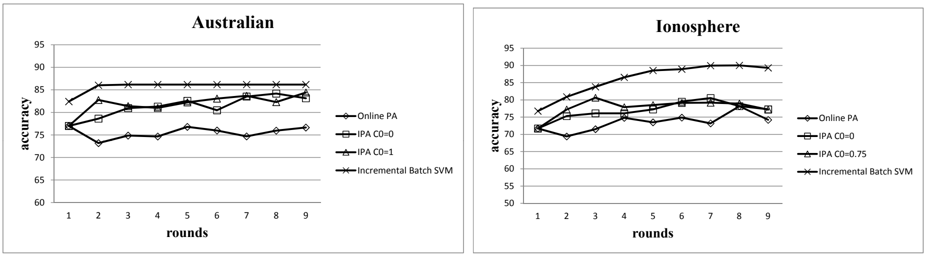


Fig. 3. Classification results of incremental learning approaches on *Australian* and *Ionosphere*

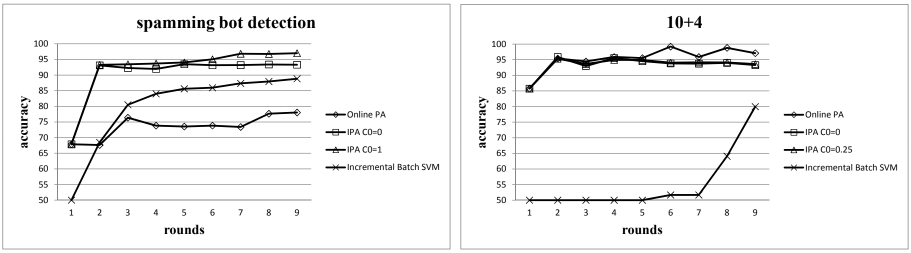


Fig. 4. Classification results of incremental learning approaches on *bot* and *10+4*

that our approach has the ability to adapt the diverse sample distribution for classifiers because we obtain better performance in accuracy than the SVM approach of which support vectors are maintained as informative samples. Mention to the performance on *Australian* and *Ionosphere*, it seems ambiguous or noise samples exist so that the approaches (PA and IPA) to incrementally update the classifier by one sample do not have impressive results. In this case, collected samples in the set might be simultaneously used for updating classifiers, like the incremental batch SVM, to filter out misleading or noise samples.

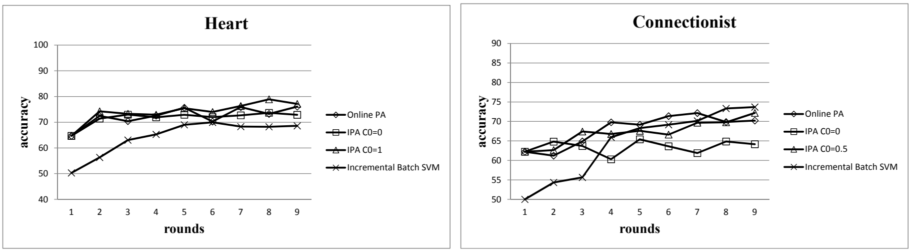


Fig. 5. Classification results of incremental learning approaches on *heart* and *Connectionist*

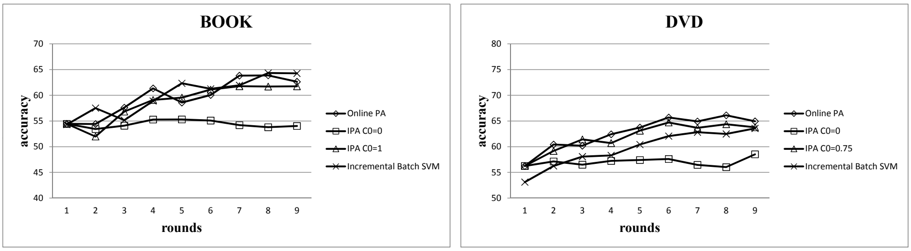


Fig. 6. Classification results of incremental learning approaches on *BOOK* and *DVD*

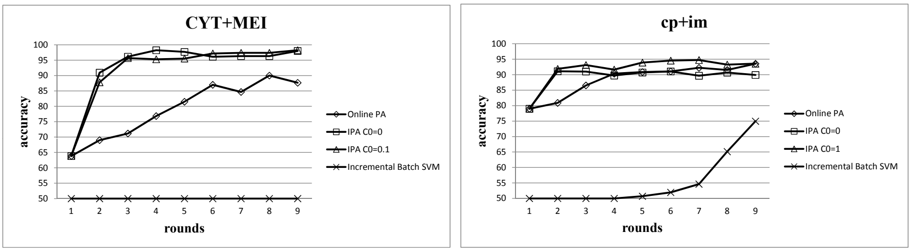


Fig. 7. Classification results of incremental learning approaches on *CYT+ME1* and *cp+im*

Interestingly on *CYT+MEI*, *cp+im*, *German*, and *Emails* in Fig. 7-8, the incremental batch SVM approach has biased results. It is observed that, in estimating performance of the classifier, it focuses on non-weighting estimated errors, instead of average weights for errors on two respective classes. Still on those datasets, proposed IPA has the practical ability to obtain the best classification accuracy. Hence, our approach to update classifiers is not affected by biased classification results.

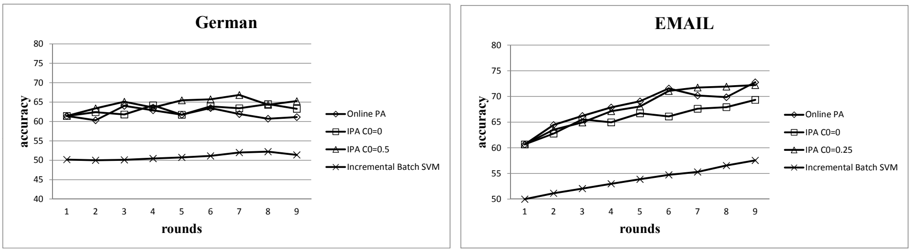


Fig. 8. Classification results of incremental learning approaches on *German* and *Emails*

5 Conclusion

In this paper, we propose an efficient incremental learning approach to deal with the practical requirement of frequently updating classifiers. Our approach is proposed to adjust the classifier incrementally using one sample in each collected set. That is, the classifier is aggressively updated by adjusting more than the margin of a sample, and its prediction mistakes are corrected as more as possible. For each potential update step, we get a closed form solution for the updated classifier through solving a simple constrained optimization problem. At last the selected classifier shall have the least prediction errors on the collected dataset. Our experimental results are presented that, when updating a classifier, it is effective to correct its prediction mistakes, in addition to minimizing the classifier deviation. And it is also shown that our approach has the ability to adapt the diverse sample distribution for classifiers. Except several datasets that consist of some misleading or noise samples, the classifier that is incrementally adjusted by our approach is able to gain remarkable classification accuracy. Therefore it is presented that the proposed approach is suitable to be applied for effectively adjusting the existing classifiers using periodically collected datasets.

References

1. Sena, G.G., Belzarena, P.: Early traffic classification using support vector machines. In: 5th International Latin American Networking Conference, pp. 60–66. ACM, New York (2009)
2. Robertson, W.K., Maggi, F., Kruegel, C., Vigna, G.: Effective Anomaly Detection with Scarce Training Data. In: The Network and Distributed System Security Symposium. ISOC (2010)
3. Du, H., Teng, S., Yang, M., Zhu, Q.: Intrusion Detection System Based on Improved SVM Incremental Learning. In: International Conference on Artificial Intelligence and Computational intelligence, pp. 23–28. IEEE Press (2009)
4. Utgoff, P.E.: Incremental Induction of Decision Trees. J. Machine Learning 4, 161–186 (1989)
5. Mohamed, S., Rubin, D., Marwala, T.: Incremental Learning for Classification of Protein Sequences. In: International Joint Conference on Neural Networks, pp. 19–24. IEEE Press (2007)

6. Chen, Z., Huang, L., Murphey, Y.L.: Incremental Learning for Text Document Classification. In: International Joint Conference on Neural NetWorks, pp. 2592–2597. IEEE Press (2007)
7. Ruping, S.: Incremental Learning with Support Vector Machines. In: International Conference on Data Mining, pp. 641–642. IEEE Press (2001)
8. Xiao, R., Wang, J., Zhang, F.: An Approach to Incremental SVM Learning Algorithm. In: International Conference on Tools with Artificial Intelligence, pp. 268–273. IEEE Press (2000)
9. Cauwenberghs, G., Poggio, T.: Incremental and Decremental Support Vector Machine Learning. In: Neural Information Processing Systems, vol. 13. MIT Press, Cambridge (2001)
10. Liu, Y., He, Q., Chen, Q.: Incremental Batch Learning with Support Vector Machines. In: 5th World Congress on Intelligent Control and Automation, pp. 1857–1861. IEEE Press (2004)
11. Crammer, K., Dekel, O., Keshet, J., Shwartz, S.S., Singer, Y.: Online Passive-Aggressive Algorithms. J. Machine Learning Research 7, 551–585 (2006)
12. Zhu, X.: Lazy Bagging for Classifying Imbalanced Data. In: 7th IEEE International Conference on Data Mining, pp. 763–768 (2007)
13. Freund, Y., Schapire, R.E.: Large Margin Classification Using the Perceptron Algorithm. J. Machine Learning 37, 277–296 (1999)
14. Ng, H.T., Goh, W.B., Low, K.L.: Feature selection, perceptron learning, and a usability case study for text categorization. In: International Conference on Research and Development in Information Retrieval, pp. 67–73. ACM, New York (1997)
15. Cesa-Bianchi, N., Conconi, A., Gentile, C.: A Second-Order Perceptron Algorithm. J. Computing 34(3), 640–668 (2005)
16. Wang, S., San, Y., Wang, S.: An Online Modeling Method Based on Support Vector Machine. In: International Conference on COmputer Science and Software Engineering, pp. 98–101. IEEE Press (2008)
17. Sculley, D., Wachman, G.M.: Relaxed Online SVMs for spam filtering. In: International Conference on Research and Development in Information Retrieval, pp. 415–422. ACM, New York (2007)
18. Dredze, M., Crammer, K., Pereira, F.: Confidence-Weighted Linear Classification. In: International Conference on Machine Learning, pp. 264–271. ACM, New York (2008)
19. Crammer, K., Kulesza, A., Dredze, M.: Adaptive Regularization of Weight Vectors. In: Neural Information Processing Systems. MIT Press, Cambridge (2009)
20. Lin, P., Yen, T., Fu, J., Yu, C.: Analyzing Anomalous Spamming Activities in a Campus Network. In: TANET (2011)

Two-View Online Learning

Tam T. Nguyen, Kuiyu Chang, and Siu Cheung Hui

School of Computer Engineering
Nanyang Technological University
50 Nanyang Avenue, Singapore 639798

Abstract. We propose a two-view online learning algorithm that utilizes two different views of the same data to achieve something that is greater than the sum of its parts. Our algorithm is an extension of the single-view Passive Aggressive (PA) algorithm, where we minimize the changes in the two view weights and disagreements between the two classifiers. The final classifier is an equally weighted sum of the individual classifiers. As a result, disagreements between the two views are tolerated as long as the final combined classifier output is not compromised. Our approach thus allows the stronger voice (view) to dominate whenever the two views disagree. This additional allowance of diversity between the two views is what gives our approach the edge, as espoused by classical ensemble learning theory. Our algorithm is evaluated and compared to the original PA algorithm on three datasets. The experimental results show that it consistently outperforms the PA algorithm on individual views and concatenated view by up to 3%.

1 Introduction

In applications where large amount of data arrives in sequence, e.g., stock market prediction and email filtering, simple online learning such as Perceptron [1], second-order Perceptron [2], and Passive Aggressive (PA) [4] algorithms can be easily deployed with reasonable performance and low computational cost.

For some domains, data may originate from several different sources, also known as views. For example, a web page may have a content view comprising text contained within it, a link view expressing its relationships to other web pages, and a revision view that tracks the different changes that it has undergone.

When the various data sources are independent, running several instances of the same algorithm on it and combining the output via an ensemble learning framework works well. A simple concatenation of the two sources in a vector space model could unnecessarily favor sources with larger number of dimensions. On the other hand, training a separate model on each source fails to make good use of the relationship among the sources, even for a baseline ensemble classifier.

To take advantage of data with multiple views, various methods such as SVM-2K [7] and alternatives [9] have been proposed. However, the two-view methods proposed so far utilizes support vector machine (SVM) [3], which is fundamentally a batch learning algorithm that cannot be easily tailored to work well on large scale online streaming data.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 74–85, 2012.

One simple approach to extend the online learning model to handle two view data is to train one model for each view independently, and combine the classifier outputs just like in classical ensemble learning. However, this approach ignores the relationship between the two views. Instead of using the same idea as SVM-2K where data in one view is used to improve the SVM performance [3] on another view (single view), we take advantage of the relationship between the two views to improve the combined performance. Specifically, we propose a novel online learning algorithm based on the PA algorithm, called Two-view Passive Aggressive (Two-view PA) learning. Our approach minimizes the difference between the two classifier outputs, but allows the outputs to differ as long as the weighted sum of each output leads to the correct result. In classical ensemble learning, the more diverse the classifier, the better the combined performance. In a way, the Two-view PA can be viewed as an ensemble of two online classifiers, except that the two views are jointly optimized.

2 Related Work

Online learning has been researched for more than 50 years. Back in 1962, Block proposed the seminal Perceptron [1] algorithm, while Novikoff [11] later provided theoretical findings, which started the first wave of Artificial Intelligence research in the mid twentieth century. The Perceptron is known to be one of the fastest online learning algorithms. However, its performance is still far from satisfactory in practice. Recently in 2005, Cesa-Bianchi et al. [2] proposed the Second-order Perceptron (SOP) algorithm, which takes advantage of second-order data to improve the accuracy of the original Perceptron. Compared with Perceptron, SOP works better in terms of accuracy but requires more time to train.

In 2006, Crammer et al. [4] proposed another Perceptron-based algorithm, namely the Passive Aggressive (PA) algorithm, which incorporates the margin maximizing criterion of modern machine learning algorithms. They not only have better performance than that of the SOP algorithm but also run significantly faster. Moreover, algorithms that improved upon the PA algorithm include the Passive-Aggressive Mahalanobis [10], the Confidence-Weight (CW) Linear Classifier [6], and its latest version, multi-class CW [5]. The CW algorithm updates its weight by minimizing the Kullback-Leibler divergence between the new and old weights. However, similar to the SOP algorithm, these algorithms are time consuming compared to the first-order PA.

The PA algorithm works better than the SOP in terms of both speed and accuracy. However, it can only process one data stream at one time. On the other hand, in batch learning, Farquhar et al. [7] proposed a large margin two-view Support Vector Machine (SVM) [3] algorithm called the SVM-2K, which is an extension of the well-known SVM algorithm. The two-view learning algorithm was shown to give better performance compared to the original SVM on different image datasets [7]. Thus, SVM-2K provides the inspiration for our current work.

3 Two-View Online Passive Aggressive Learning

3.1 Problem Setting

Online learning aims to learn the weight $\mathbf{w}$ of a linear prediction function $f(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x})$. The online learning algorithm operates in rounds, as input data arrives sequentially. Let $x_t \in \mathbb{R}^n$ be an example arriving at round t. The algorithm predicts its label $\hat{y}_t \in \{-1, +1\}$, after which it receives the true label. If its prediction is correct, the learning process proceeds to the next round. Otherwise, it suffers a loss $\ell(y_t, \hat{y}_t)$, and updates its weight $\mathbf{w}$ accordingly. The loss can be modeled using the hinge-loss function, which equals to zero when the margin exceeds 1, as follows.

$$\ell(\mathbf{w}_t; (\mathbf{x}_t, y_t)) = \begin{cases} 0 & \text{if} \quad y_t(\mathbf{w}_t \cdot \mathbf{x}_t) \geq 1 \\ 1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t) & \text{otherwise} \end{cases} \tag{1}$$

The overall objective is to minimize the cumulative loss over the entire sequence of examples. From this, Crammer et al. [4] formulated three optimization problems; one based on hard margin and two using soft margins, respectively named PA, PA-I, and PA-II with weight update equations as follows.

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$$

where the coefficient τ_t has one of three forms.

$$\begin{aligned} \tau_t &= \frac{1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t)}{\| \mathbf{x}_t \|^2} && \text{(PA)} \\ \tau_t &= \min\left\{C, \frac{1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t)}{\| \mathbf{x}_t \|^2}\right\} && \text{(PA-I)} \\ \tau_t &= \frac{1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t)}{\| \mathbf{x}_t \|^2 + \frac{1}{2C}} && \text{(PA-II)} \end{aligned}$$

The performance of the soft margin based PA-I and PA-II algorithms are almost identical, and both performed better than the hard margin based PA algorithm [4]. Therefore, in this work, our proposed algorithm will be developed based on the PA-I algorithm.

For the two-view online learning setting, training data are triplets $(\mathbf{x}_t^A, \mathbf{x}_t^B, y_t) \in \mathbb{R}^n \times \mathbb{R}^m \times [-1, +1]$, which arrives in sequence where $\mathbf{x}_t^A \in \mathbb{R}^n$ is the first view vector, $\mathbf{x}_t^B \in \mathbb{R}^m$ is the second view vector, and y_t is their common label. The goal is to learn the coupled weights $(\mathbf{w}_t^A, \mathbf{w}_t^B)$ of a *hybrid model* defined as follows.

$$f(\mathbf{x}_t^A, \mathbf{x}_t^B) = \text{sign}\frac{1}{2}(\mathbf{w}_t^A \cdot \mathbf{x}_t^A + \mathbf{w}_t^B \cdot \mathbf{x}_t^B)$$

To incorporate the hybrid classifier, we modify the loss function as follows.

$$\ell((\mathbf{w}_t^A,\mathbf{w}_t^B);(\mathbf{x}_t^A,\mathbf{x}_t^B,y_t)) = \begin{cases} 0 & \text{if } \frac{1}{2}y_t(\mathbf{w}_t^A\cdot\mathbf{x}_t^A+\mathbf{w}_t^B\cdot\mathbf{x}_t^B)\geq 1 \\ 1-\frac{1}{2}y_t(\mathbf{w}_t^A\cdot\mathbf{x}_t^A+\mathbf{w}_t^B\cdot\mathbf{x}_t^B) & \text{otherwise} \end{cases} \tag{2}$$

3.2 Relationship between Views

The primary challenge of multi-view learning is to properly define the relatedness among the different views. In other words, the relatedness quantifies the agreement among the views. Moreover, one could simply disregard the agreement between the two prediction functions, but instead learn the hybrid prediction function. Specifically, we want the hybrid prediction function $f(\mathbf{x}_t^A,\mathbf{x}_t^B) = \text{sign}\frac{1}{2}(\mathbf{w}_t^A\cdot\mathbf{x}_t^A+\mathbf{w}_t^B\cdot\mathbf{x}_t^B)$ to optimally predict the correct labels of examples. In this case, we do not really care whether $f(\mathbf{x}_t^A)$ or $f(\mathbf{x}_t^B)$ can individually classify the example correctly; what we want is for their equally weighted sum $f(\mathbf{x}_t^A,\mathbf{x}_t^B)$ to correctly predict the class label.

Generally, we want the two views to agree with one another. This can be enforced by minimizing their L1-norm or L2-norm disagreements as follows.

$$\sum_{t=1}^{T}|\mathbf{w}_t^A\cdot\mathbf{x}_t^A-\mathbf{w}_t^B\cdot\mathbf{x}_t^B| \quad \text{or} \quad \sum_{t=1}^{T}(\mathbf{w}_t^A\cdot\mathbf{x}_t^A-\mathbf{w}_t^B\cdot\mathbf{x}_t^B)^2 \tag{3}$$

where $|\cdot|$ denotes the absolute function. Here we use L1-norm instead of L2-norm because it is harder to find a close-form solution for the latter. In the next section, we will define an optimization problem based on the L1-norm relatedness measure.

3.3 Two-View Passive Aggressive Algorithm

The ideal objective function should include both the new loss function in (2) and the view relatedness function in (3). Similar to the PA algorithm, the new weights of the two-view learning algorithm are updated based on the optimization problem as follows.

$$\begin{aligned} (\mathbf{w}_{t+1}^A,\mathbf{w}_{t+1}^B) = & \underset{(\mathbf{w}^A,\mathbf{w}^B)\in\mathbb{R}^n\times\mathbb{R}^m}{\text{argmin}} \frac{1}{2}\parallel\mathbf{w}^A-\mathbf{w}_t^A\parallel^2+\frac{1}{2}\parallel\mathbf{w}^B-\mathbf{w}_t^B\parallel^2 \\ & +\gamma|y_t\mathbf{w}^A\cdot\mathbf{x}_t^A-y_t\mathbf{w}^B\cdot\mathbf{x}_t^B|+C\xi \\ \text{s.t.} \quad & 1-\frac{1}{2}(y_t\mathbf{w}^A\cdot\mathbf{x}_t^A+y_t\mathbf{w}^B\cdot\mathbf{x}_t^B)\leq\xi;\xi\geq 0 \end{aligned}$$

where γ and C are positive agreement and aggressiveness parameters respectively. While γ is used to adjust the importance of the agreement between the two views, C is used to control the aggressiveness property of the PA algorithm. Note that the y_t multiplier in the agreement is there just for subsequent derivation convenience.

For the absolute function, we have

$$|y_t\mathbf{w}^A \cdot \mathbf{x}_t^A - y_t\mathbf{w}^B \cdot \mathbf{x}_t^B| = \max(y_t\mathbf{w}^A \cdot \mathbf{x}_t^A - y_t\mathbf{w}^B \cdot \mathbf{x}_t^B, y_t\mathbf{w}^B \cdot \mathbf{x}_t^B - y_t\mathbf{w}^A \cdot \mathbf{x}_t^A)$$

Suppose $z = |y_t\mathbf{w}^A \cdot \mathbf{x}_t^A - y_t\mathbf{w}^B \cdot \mathbf{x}_t^B|$, the above optimization problem can be expressed as follows.

$$\begin{aligned}(\mathbf{w}_{t+1}^A, \mathbf{w}_{t+1}^B) = \underset{(\mathbf{w}^A,\mathbf{w}^B)\in\mathbb{R}^n\times\mathbb{R}^m}{\operatorname{argmin}} & \frac{1}{2} \parallel \mathbf{w}^A - \mathbf{w}_t^A \parallel^2 + \frac{1}{2} \parallel \mathbf{w}^B - \mathbf{w}_t^B \parallel^2 + \gamma z + C\xi \\ \text{s.t.} \qquad & 1 - \frac{1}{2}(y_t\mathbf{w}^A \cdot \mathbf{x}_t^A + y_t\mathbf{w}^B \cdot \mathbf{x}_t^B) \leq \xi; \\ & \xi \geq 0; \\ & z \geq y_t\mathbf{w}^A \cdot \mathbf{x}_t^A - y_t\mathbf{w}^B \cdot \mathbf{x}_t^B; \\ & z \geq y_t\mathbf{w}^B \cdot \mathbf{x}_t^B - y_t\mathbf{w}^A \cdot \mathbf{x}_t^A.\end{aligned}$$

Next, we define the Lagrangian of the optimization problem as follows.

$$\begin{aligned}\mathcal{L} = & \frac{1}{2} \parallel \mathbf{w}^A - \mathbf{w}_t^A \parallel^2 + \frac{1}{2} \parallel \mathbf{w}^B - \mathbf{w}_t^B \parallel^2 + \gamma z + C\xi \\ & + \tau\Big(1 - \xi - \frac{1}{2}(y_t\mathbf{w}^A \cdot \mathbf{x}_t^A + y_t\mathbf{w}^B \cdot \mathbf{x}_t^B)\Big) - \lambda\xi \\ & + \alpha(y_t\mathbf{w}^A \cdot \mathbf{x}_t^A - y_t\mathbf{w}^B \cdot \mathbf{x}_t^B - z) + \beta(y_t\mathbf{w}^B \cdot \mathbf{x}_t^B - y_t\mathbf{w}^A \cdot \mathbf{x}_t^A - z) \qquad (4) \\ = & \frac{1}{2} \parallel \mathbf{w}^A - \mathbf{w}_t^A \parallel^2 + \frac{1}{2} \parallel \mathbf{w}^B - \mathbf{w}_t^B \parallel^2 + (\gamma - \alpha - \beta)z + (C - \lambda - \tau)\xi \\ & + (\alpha - \beta - \frac{1}{2}\tau)y_t\mathbf{w}^A \cdot \mathbf{x}_t^A + (\beta - \alpha - \frac{1}{2}\tau)y_t\mathbf{w}^B \cdot \mathbf{x}_t^B + \tau\end{aligned}$$

where α, β, τ, and λ are positive Lagrangian multipliers.

Setting the partial derivatives of $\mathcal{L}$ with respect to the weight $\mathbf{w}^A$ to zero, we have,

$$0 = \frac{\partial \mathcal{L}}{\partial \mathbf{w}^A} = \mathbf{w}^A - \mathbf{w}_t^A + (\alpha - \beta - \frac{1}{2}\tau)y_t\mathbf{x}_t^A \Rightarrow \mathbf{w}^A = \mathbf{w}_t^A - (\alpha - \beta - \frac{1}{2}\tau)y_t\mathbf{x}_t^A \quad (5)$$

Similarly, for the other view we have

$$\mathbf{w}^B = \mathbf{w}_t^B - (\beta - \alpha - \frac{1}{2}\tau)y_t\mathbf{x}_t^B \quad (6)$$

Setting the partial derivatives of $\mathcal{L}$ with respect to weight z to zero, we have

$$0 = \frac{\partial \mathcal{L}}{\partial z} = (\gamma - \alpha - \beta) \Rightarrow \alpha + \beta = \gamma \quad (7)$$

Setting the partial derivatives of $\mathcal{L}$ with respect to weight ξ to zero, we have,

$$0 = \frac{\partial \mathcal{L}}{\partial \xi} = (C - \lambda - \tau) \Rightarrow \lambda + \tau = C \quad (8)$$

Note that $\lambda \geq 0$, thus we can conclude that $0 \leq \tau \leq C$.

Substituting (5), (6), (7), and (8) into (4), we have,

$$\begin{aligned}\mathcal{L} &= \frac{1}{2}(\alpha-\beta-\frac{1}{2}\tau)^2 \parallel \mathbf{x}_t^A \parallel^2 + \frac{1}{2}(\beta-\alpha-\frac{1}{2}\tau)^2 \parallel \mathbf{x}_t^B \parallel^2 \\ &\quad +(\alpha-\beta-\frac{1}{2}\tau)y_t\Big(\mathbf{w}_t^A-(\alpha-\beta-\frac{1}{2}\tau)y_t\mathbf{x}_t^A\Big)\mathbf{x}_t^A \\ &\quad +(\beta-\alpha-\frac{1}{2}\tau)y_t\Big(\mathbf{w}_t^B-(\alpha-\beta-\frac{1}{2}\tau)y_t\mathbf{x}_t^B\Big)\mathbf{x}_t^B+\tau \\ &= -\frac{1}{2}(\alpha-\beta-\frac{1}{2}\tau)^2 \parallel \mathbf{x}_t^A \parallel^2 - \frac{1}{2}(\beta-\alpha-\frac{1}{2}\tau)^2 \parallel \mathbf{x}_t^B \parallel^2 \\ &\quad +(\alpha-\beta-\frac{1}{2}\tau)y_t\mathbf{w}_t^A\cdot\mathbf{x}_t^A+(\beta-\alpha-\frac{1}{2}\tau)y_t\mathbf{w}_t^B\cdot\mathbf{x}_t^B+\tau\end{aligned} \tag{9}$$

Setting the partial derivatives of $\mathcal{L}$ with respect to weight τ to zero, we have,

$$\begin{aligned}0=\frac{\partial\mathcal{L}}{\partial\tau} &= \frac{1}{2}(\alpha-\beta-\frac{1}{2}\tau)\parallel \mathbf{x}_t^A \parallel^2 + \frac{1}{2}(\beta-\alpha-\frac{1}{2}\tau)\parallel \mathbf{x}_t^B \parallel^2 \\ &\quad +1-\frac{1}{2}(y_t\mathbf{w}_t^A\cdot\mathbf{x}_t^A+y_t\mathbf{w}_t^B\cdot\mathbf{x}_t^B)\end{aligned}$$

$$\Rightarrow \tau = \frac{2}{\parallel \mathbf{x}_t^A \parallel^2 + \parallel \mathbf{x}_t^B \parallel^2}\Big((\alpha-\beta)(\parallel \mathbf{x}_t^A \parallel^2 - \parallel \mathbf{x}_t^B \parallel^2)+2\ell_t\Big)$$

where the loss $\ell_t = 1-\frac{1}{2}(y_t\mathbf{w}_t^A\cdot\mathbf{x}_t^A+y_t\mathbf{w}_t^B\cdot\mathbf{x}_t^B)$. For the sake of simplicity, we denote,

$$a=\frac{2}{\parallel \mathbf{x}_t^A \parallel^2 + \parallel \mathbf{x}_t^B \parallel^2} \qquad \text{and} \qquad b=\parallel \mathbf{x}_t^A \parallel^2 \parallel \mathbf{x}_t^B \parallel^2 \tag{10}$$

As mentioned in Equation (8), we have $\tau+\lambda=C$ and $\lambda\geq 0$ so we can conclude that $\tau\leq C$. Now τ can be determined as follows:

$$\tau=\min\Big\{C, a\Big((\alpha-\beta)(\parallel \mathbf{x}_t^A \parallel^2 - \parallel \mathbf{x}_t^B \parallel^2)+2\ell_t\Big)\Big\} \tag{11}$$

Substituting (11) into (9), we have,

$$\begin{aligned}\mathcal{L} &= -\frac{1}{2}a\Big((\alpha-\beta)\parallel \mathbf{x}_t^B \parallel^2-\ell_t\Big)^2 \parallel \mathbf{x}_t^A \parallel^2 - \frac{1}{2}a\Big((\beta-\alpha)\parallel \mathbf{x}_t^A \parallel^2-\ell_t\Big)^2 \parallel \mathbf{x}_t^B \parallel^2 \\ &\quad +a((\alpha-\beta)\parallel \mathbf{x}_t^B \parallel^2-\ell_t)y_t\mathbf{w}_t^A\cdot\mathbf{x}_t^A+a((\beta-\alpha)\parallel \mathbf{x}_t^A \parallel^2-\ell_t)y_t\mathbf{w}_t^B\cdot\mathbf{x}_t^B \\ &\quad +a\Big((\alpha-\beta)(\parallel \mathbf{x}_t^A \parallel^2 - \parallel \mathbf{x}_t^B \parallel^2)+2\ell_t\Big)\end{aligned} \tag{12}$$

Setting the partial derivatives of $\mathcal{L}$ with respect to weight α to zero, we have,

$$\begin{aligned}0=\frac{\partial\mathcal{L}}{\partial\alpha} &= a\Big((\alpha-\beta)\parallel \mathbf{x}_t^B \parallel^2+\ell_t\Big)b+a\Big((\beta-\alpha)\parallel \mathbf{x}_t^A \parallel^2+\ell_t\Big)b \\ &\quad +a(\parallel \mathbf{x}_t^B \parallel^2 y_t\mathbf{w}_t^A\cdot\mathbf{x}_t^A - \parallel \mathbf{x}_t^A \parallel^2 y_t\mathbf{w}_t^B\cdot\mathbf{x}_t^B + \parallel \mathbf{x}_t^A \parallel^2 - \parallel \mathbf{x}_t^B \parallel^2) \\ &= a\Big((\alpha-\beta)\parallel \mathbf{x}_t^B \parallel^2+\ell_t)\Big)b+a\Big((\beta-\alpha)\parallel \mathbf{x}_t^A \parallel^2+\ell_t)\Big)b \\ &\quad +a(\parallel \mathbf{x}_t^A \parallel^2 \ell_t^B - \parallel \mathbf{x}_t^B \parallel^2 \ell_t^A)\end{aligned}$$

where $\ell_t^A = 1-y_t\mathbf{w}_t^A\cdot\mathbf{x}_t^A$ and $\ell_t^B=1-y_t\mathbf{w}_t^B\cdot\mathbf{x}_t^B$. We also have $\alpha+\beta=\gamma$. Therefore, we can conclude that

$$\alpha = \frac{\gamma}{2} + \frac{1}{2}\frac{1}{\| \mathbf{x}_t^A \|^2 + \| \mathbf{x}_t^B \|^2}\Big(\frac{\ell_t^B}{\| \mathbf{x}_t^B \|^2} - \frac{\ell_t^A}{\| \mathbf{x}_t^A \|^2}\Big) \tag{13}$$

Similarly, we have

$$\beta = \frac{\gamma}{2} - \frac{1}{2}\frac{1}{\| \mathbf{x}_t^A \|^2 + \| \mathbf{x}_t^B \|^2}\Big(\frac{\ell_t^B}{\| \mathbf{x}_t^B \|^2} - \frac{\ell_t^A}{\| \mathbf{x}_t^A \|^2}\Big) \tag{14}$$

Recall that we have $\alpha \geq 0$, $\beta \geq 0$, and $\alpha+\beta = \gamma$. Hence, we can conclude that $\alpha \leq \gamma$ and $\beta \leq \gamma$. Finally, we obtain our Two-view Passive Aggressive formulation as shown in Algorithm 1. The optimal value of the two tuning parameters C and γ can be estimated via cross validation in practice.

Algorithm 1. Two-view Passive Aggressive Algorithm

```
Input:
  $C$ = positive aggressiveness parameter
  $\gamma$ = positive agreement parameter
Output:
  None
Process:
Initialize $\mathbf{w}_1^A \leftarrow \mathbf{0}$; $\mathbf{w}_1^B \leftarrow \mathbf{0}$;
for $t = 1, 2, \dots$ do
    Receive instances $\mathbf{x}_t^A \in \mathbb{R}^n$ and $\mathbf{x}_t^B \in \mathbb{R}^m$
    Predict $\hat{y}_t = \text{sign}\frac{1}{2}(\mathbf{w}_t^A \cdot \mathbf{x}_t^A + \mathbf{w}_t^B \cdot \mathbf{x}_t^B)$
    Receive correct label $y_t \in \{-1, +1\}$
    Suffer loss $\ell_t \leftarrow \max\Big\{0, 1 - y_t\frac{1}{2}(\mathbf{w}_t^A \cdot \mathbf{x}_t^A + \mathbf{w}_t^B \cdot \mathbf{x}_t^B)\Big\}$
    if $\ell_t > 0$ then
        Set $\ell_t^A \leftarrow 1 - y_t\mathbf{w}_t^A \cdot \mathbf{x}_t^A$; $\ell_t^B \leftarrow 1 - y_t\mathbf{w}_t^B \cdot \mathbf{x}_t^B$
            $\alpha \leftarrow \max\Big\{0, \min\{\gamma, \frac{1}{2}\Big(\gamma + \frac{\frac{\ell_t^B}{\|\mathbf{x}_t^B\|^2} - \frac{\ell_t^A}{\|\mathbf{x}_t^A\|^2}}{\| \mathbf{x}_t^A \|^2 + \| \mathbf{x}_t^B \|^2}\Big)\}\Big\}$
            $\beta \leftarrow \max\Big\{0, \min\{\gamma, \frac{1}{2}\Big(\gamma - \frac{\frac{\ell_t^B}{\|\mathbf{x}_t^B\|^2} - \frac{\ell_t^A}{\|\mathbf{x}_t^A\|^2}}{\| \mathbf{x}_t^A \|^2 + \| \mathbf{x}_t^B \|^2}\Big)\}\Big\}$
            $\tau_t \leftarrow \min\Big\{C, \frac{(\alpha - \beta)(\| \mathbf{x}_t^A \|^2 - \| \mathbf{x}_t^B \|^2) + 2\ell_t}{\| \mathbf{x}_t^A \|^2 + \| \mathbf{x}_t^B \|^2}\Big\}$
        Update $\mathbf{w}_{t+1}^A \leftarrow \mathbf{w}_t^A - (\alpha - \beta - \frac{1}{2}\tau_t)y_t\mathbf{x}_t^A$
               $\mathbf{w}_{t+1}^B \leftarrow \mathbf{w}_t^B - (\beta - \alpha - \frac{1}{2}\tau_t)y_t\mathbf{x}_t^B$
    end
end
```

4 Performance Evaluation

In this section, we evaluate the online classification performance of our proposed Two-view PA on 3 benchmark datasets, Ads [8], Product Review [9], and WebKB [12]). The *single-view* PA algorithm serves as the baseline. We use a different PA model for each view, naming them *PA View 1* and *PA View 2*. We also concatenate the input feature vectors from each view to form a larger feature set, and report the results. We denote this alternative approach as *PA Cat*. The dataset summary statistics are shown in Table 1. We note that the Ads and WebKB datasets are very imbalanced, which led us to use F-measure instead of accuracy to evaluate the classification performance. To be fair, we choose $C = 0.1$ and $\gamma = 0.5$ for all PA algorithms. All experiments were conducted using 5-fold cross validation.

Table 1. Summary statistics of 3 datasets

	View		Sample Count		
	Name	#Dimension	#Positive	#Negative	#Total
Ads	img & dest url	929	459	2820	3279
	alt & base url	602			
WebKB	page	3000	230	821	1051
	link	1840			
Product Review	lexical	2759	1000	1000	2000
	formal	5			

4.1 View Difference Comparison

At round t, the view difference is defined as $|\mathbf{w}_t^A \cdot \mathbf{x}_t^A - \mathbf{w}_t^B \cdot \mathbf{x}_t^B|$, which shows the difference in prediction output between the two views. Figures 1(a), 2(a), and 3(a) show the view differences for the three datasets, respectively.

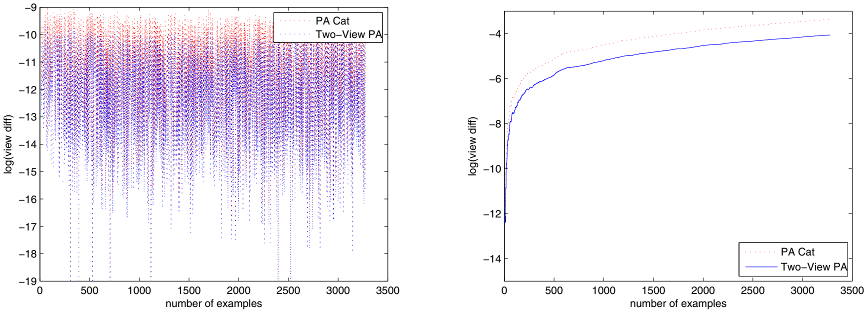
Figures 1(b), 2(b), and 3(b) plot the cumulative view difference at round t, $\frac{1}{T}\sum_{t=1}^{T} |\mathbf{w}_t^A \cdot \mathbf{x}_t^A - \mathbf{w}_t^B \cdot \mathbf{x}_t^B|$. This measures the relationship between the two views as the algorithm adapts to the dataset. The smaller it is, the more related the two views.

Compared to the Product Review and WebKB datasets, the view difference for the Ads dataset varies very much. This means that the agreement between the two views is not stable. As expected, its cumulative view difference turns out to be the largest among the three datasets. Hence, we would expect a classifier based on simple concatenation of the two views to yield poor classification performance. This is in fact confirmed subsequently in the poor PA Cat result for the Ads dataset in Table 2.

On the other end of the spectrum, both the average and cumulative view difference for the WebKB dataset is the smallest. Therefore, one should be able to combine the two views into a single view and just run a simple PA algorithm to obtain a decent classification performance. This hypothesis is confirmed in Table 2, where the PA Cat result outperforms either view by more than 2%.

4.2 Ads Dataset

The Ads dataset was first used by Kushmerick [8] to automatically filter advertisement images from web pages. The Ads dataset comprises more than two views. In this experiment, we only use four views including *image URL view*, *destination URL view*, *base URL view*, and *alt view*. The first and second original views were concatenated as View 1 and the remaining two original views were concatenated as View 2. This dataset has 3279 examples, including 459 positive examples (ads), with the remaining as negative examples (non-ads).



(a) View Difference (b) Cumulative View Difference

Fig. 1. View Difference of the Ads Dataset

Table 2. F1 measure on 3 datasets

Dataset	PA View 1	PA View 2	PA Cat	Two-view PA
Ads	83.69 ± 3.04	76.01 ± 2.88	81.08 ± 1.99	**85.74** ± 1.97
Product Review	86.46 ± 4.59	69.20 ± 5.20	86.87 ± 3.99	**88.54** ± 1.85
WebKB	92.83 ± 1.72	92.71 ± 3.66	94.97 ± 1.80	**97.50** ± 1.80

The experimental results on the Ads dataset are shown in Table 2, where the F-measure of the proposed algorithm is the best. The Two-view PA performed up to 2% better than the runner-up, PA View 1. As previously discussed, PA View 1 is better than PA Cat since the two views have quite different classification outputs.

4.3 Product Review Dataset

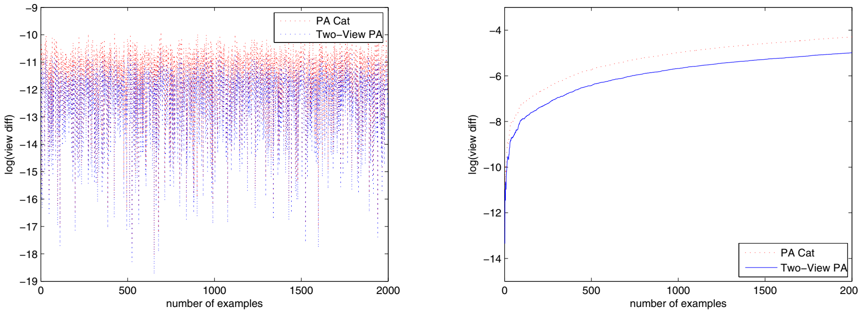
The Product Review dataset is crawled from popular online Chinese cell-phone forums [9]. The dataset has 1000 true reviews and 1000 spam reviews. It consists of two sets of features: one based on review content (*lexical view*) and the other based on extracted characteristics of the review sentences (*formal view*).

The experimental results on this dataset are shown in Table 2. Again, Two-view PA performs better than the other algorithms. The improvement is more than 2% compared with the runner-up. In this dataset, PA Cat performed better than either view alone. This is expected since the view difference between the two views are quite small, as shown in Figure 2.

Moreover, PA Cat is only 0.41% better than the best individual PA View 1. This is because PA Cat does not take into account the view relatedness information. The best performer here is the Two-view PA, which beats the runner-up by almost 2%.

4.4 WebKB Course Dataset

The WebKB course dataset has been frequently used in the empirical study of multi-view learning. It comprises 1051 web pages collected from the computer science departments of four universities. Each page has a class label, course or non-course. The two views of each page are the textual content of a web page (*page view*) and the words that occur in the hyperlinks of other web pages pointing to it (*link view*), respectively. We used a processed version of the WebKB course dataset [12] in our experiment.

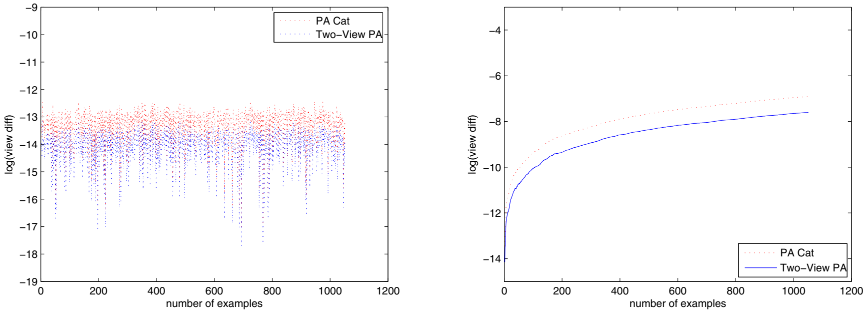


(a) View Difference (b) Cumulative View Difference

Fig. 2. View Difference of the Product Review Dataset

The performance of PA Cat here is also better than the best single view PA. However, the view difference of Two-view PA is much smaller than that of the PA algorithm as shown in Figure 3. Hence, Two-view PA performed more than 3% better than PA Cat, and 5% better than the best individual view PA.

Compared to the Ads and Product Review datasets, the view difference on the WebKB dataset is the smallest. It means that we are able to combine the two views into a single view. Therefore, the PA Cat performance on the WebKB dataset is improved more than 2% compared with the individual view PA.



(a) View Difference (b) Cumulative View Difference

Fig. 3. View Difference of the WebKB Dataset

5 Conclusion and Open Problems

In this paper, we proposed a hybrid model for two-view passive aggressive algorithm, which is able to take advantage of multiple views of data to achieve an improvement in overall classification performance. We formulate our learning framework into an optimization problem and derive a closed form solution.

There remain some interesting open problems that warrant further investigation. For one, at each round we could adjust the weight of each view so that the better view dominates. In the worst case where the two views are completely related or co-linear, e.g., view 1 is equal to view 2, our Two-view PA degenerates nicely into a single view PA. We would also like to extend Two-view PA to handle multiple views and multiple classes. Formulating a multi-view PA is nontrivial, as it involves defining multi-view relatedness and minimizing (V choose 2) view agreements, for a V-view problem. Formulating a multi-class Two-view PA should be more feasible.

References

1. Block, H.: The perceptron: A model for brain functioning. Rev. Modern Phys. 34, 123–135 (1962)
2. Cesa-Bianchi, N., Conconi, A., Gentile, C.: A second-order perceptron algorithm. Siam J. of Comm. 34 (2005)
3. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20, 273–297 (1995)
4. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. Journal of Machine Learning Research, 551–585 (2006)
5. Crammer, K., Dredze, M., Kulesza, A.: Multi-class confidence weighted algorithms. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pp. 496–504. Association for Computational Linguistics, Singapore (2009)

6. Dredze, M., Crammer, K., Pereira, F.: Confidence-weighted linear classification. In: ICML 2008: Proceedings of the 25th International Conference on Machine Learning, pp. 264–271. ACM, New York (2008)
7. Farquhar, J.D.R., Hardoon, D.R., Meng, H., Shawe-Taylor, J., Szedmák, S.: Two view learning: Svm-2k, theory and practice. In: Proceedings of NIPS 2005 (2005)
8. Kushmerick, N.: Learning to remove internet advertisements. In: Proceedings of the Third Annual Conference on Autonomous Agents, AGENTS 1999, pp. 175–181. ACM, New York (1999)
9. Li, G., Hoi, S.C.H., Chang, K.: Two-view transductive support vector machines. In: Proceedings of SDM 2010, pp. 235–244 (2010)
10. Nguyen, T.T., Chang, K., Hui, S.C.: Distribution-aware online classifiers. In: Walsh, T. (ed.) IJCAI, pp. 1427–1432. IJCAI/AAAI (2011)
11. Novikoff, A.: On convergence proofs of perceptrons. In: Proceedings of the Symposium on the Mathematical Theory of Automata, vol. 7, pp. 615–622 (1962)
12. Sindhwani, V., Niyogi, P., Belkin, M.: Beyond the point cloud: from transductive to semi-supervised learning. In: Proceedings of the 22nd International Conference on Machine Learning, ICML 2005, pp. 824–831. ACM, New York (2005)

A Generic Classifier-Ensemble Approach for Biomedical Named Entity Recognition

Zhihua Liao[1] and Zili Zhang[2,3,*]

[1] Modern Foreign-Language Education Technology Center, Foreign Studies College
Hunan Normal University, CS 410081, China
[2] Faculty of Computer and Information Science, Southwest University, CQ 400715, China
[3] School of Information Technology, Deakin University, VIC 3217, Australia
liao.zhihua61@gmail.com, zzhang@deakin.edu.au

Abstract. In named entity recognition (NER) for biomedical literature, approaches based on combined classifiers have demonstrated great performance improvement compared to a single (best) classifier. This is mainly owed to sufficient level of diversity exhibited among classifiers, which is a selective property of classifier set. Given a large number of classifiers, how to select different classifiers to put into a classifier-ensemble is a crucial issue of multiple classifier-ensemble design. With this observation in mind, we proposed a generic genetic classifier-ensemble method for the classifier selection in biomedical NER. Various diversity measures and majority voting are considered, and disjoint feature subsets are selected to construct individual classifiers. A basic type of individual classifier – Support Vector Machine (SVM) classifier is adopted as SVM-classifier committee. A multi-objective Genetic algorithm (GA) is employed as the classifier selector to facilitate the ensemble classifier to improve the overall sample classification accuracy. The proposed approach is tested on the benchmark dataset – GENIA version 3.02 corpus, and compared with both individual best SVM classifier and SVM-classifier ensemble algorithm as well as other machine learning methods such as CRF, HMM and MEMM. The results show that the proposed approach outperforms other classification algorithms and can be a useful method for the biomedical NER problem.

1 Introduction

With the wide applications of information technology in biomedical field, biomedical technology has developed very rapidly. This in turn produces a large amount of biomedical data such as human gene bank. Consequently, biomedical literature available from the Web has experienced unprecedented growth over the past few years. The amount of literature in *MEDLINE* grows by nearly 400,000 citations each year. To mine information from the biomedical databases, a helpful and useful pre-processing step is to extract the valuable biomedical named entity. In other words, this step needs to identify some names from scientific text that is not structured as traditional databases and classify these different names. As a result, biomedical named entity recognition (BioNER) becomes one of the most important issues in automatic text extraction system. Many

[*] Corresponding author.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 86–97, 2012.

popular classification algorithms have been applied to this bioNER problem. These algorithms include Support Vector Machine (SVM) [1,18,19], Conditional Random Fields (CRFs) [3], the Hidden Markov Model (HMM) [5], the Maximum Entropy (ME) [15], decision tree [16], and so on. While successful, each classifier has its own shortcomings and none of them could consistently perform well over all different datasets. To overcome the shortcomings of individual methods, ensemble method has been suggested as a promising alternative.

Ensemble method is more attractive than individual classification algorithm in that it is an effective approach for improving the prediction accuracy of a single classification algorithm. An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to classify new examples [8,11]. One of the most active areas of research in supervised learning has been to study methods for constructing good ensembles of classifiers. The most important property of successful ensemble methods is if the individual classifiers have error rate below 0.5 when classifying sample data while these errors are uncorrelated at least in some extent. That is, a necessary and sufficient condition for an ensemble of classifiers over its individual members is that the classifiers are accurate and diverse. Several recent studies indicate that the ensemble learning could improve the performance of a single classifier in many real world text classification [6,7,9,10,12,13,14,23,24].

In this paper, we propose a generic genetic classifier-ensemble approach, which employs multi-objective genetic algorithm and SVM based classifiers to construct an ensemble classifier. Each SVM based classifier is trained on a different feature subset and used as the classification committee. The rest of the paper is organized as follows: Section 2 discusses the generic genetic classifier-ensemble approach in detail. Experimental results and analysis are provided in Section 3. Conclusions and future work are presented in Section 4.

2 The Generic Genetic Classifier-Ensemble Approach

Classifier-ensemble is a popular technique in pattern recognition domain. It reflects the generalization accuracy if an ensemble depends not only on the performances of the individual classifier but also on the diversity among the classifiers [6,8,10,7,12,22]. Therefore, a classifier-ensemble system is usually made up of two major components: the classifiers forming the ensemble members and the combination scheme. In order to achieve this goal, we develop a generic genetic classifier-ensemble algorithm. In the proposed approach, SVM is used as the basic classifier and the genetic algorithm was used to search the optimal solution of weighted classifier combination.

2.1 Feature Set and SVM Based Classifier

Since the main issue using machine learning method for BioNER task is to design a proper feature set, choosing the suitable feature is very important for improving the performance of the system. Here various types of features have been considered for bioNER task in different combinations (see Table 1).

- Word: All words appearing in the training data.
- Orthography: Table 2 shows the orthographic features. If the token has more than one feature, then we used the feature list of Table 2 from left to right and from up to down orderly.
- Prefix: Uni-,bi-, and tri-grams(in letters) of the starting letters of the current token.
- Suffix: Uni-,bi-, and tri-grams(in letters) of the ending letters of the current token.
- Lexical: POS tags, base phrase classes, and base noun phrase chunks. POS tags are generated by Geniatagger[1].
- Preceding class: The prediction of the classifier for the preceding tokens are computed dynamically and used as feature.
- Surface word: Surface words forming a list of tokens that are tagged as an entity in the training data. In our system, the surface word includes simple surface word lists, name aliases and trigger words [17,21].

Table 1. The features in our generic genetic classifier-ensemble system

Feature	Value
words	all words in the training data
orthographic	capital, symbol, etc.(see Table 2)
prefix	1,2, and 3 gram of starting letters of word
suffix	1,2, and 3 gram of ending letters of word
lexical	POS tags, base phrase classes, and base noun phrase chunks
preceding class	-4,-3, -2, -1
surface word	simple surface word lists, name aliases and trigger words

Table 2. Orthographic features

Feature	Example	Feature	Example
DigitNumber	15	Greek	alpha
SingleCap	M	CapsAndDigits	I2
TwoCaps	RalGDS	LettersAndDigits	p52
InitCaps	Interleukin	LowCaps	kappaB
Lowercase	kinases	Hyphen	-
Backslash	/	OpenSquare	[
CloseSquare	]	Colon	:
SemiColon	;	Pecent	%
OpenParen	(	CloseParen	)
Comma	,	FullStop	.
Determiner	the	Conjunction	and
Other	* @		

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/

Table 3. The parameters of Yamcha

Parameter	Value
kernel	polynomial
degree of kernel	1,2,3
direction of parsing	forward, backward
windows position	9 words(position -4, -3,-2,-1,0,+1,+2,+3,+4)
multi-class	pair-wise

Next, due to the fact that support vector machines(SVMs) are powerful methods for learning a classifier and have been applied successfully to many NLP tasks, SVMs construct the base classifier in BioNER. The general-purpose text chunker named Yet Another Multipurpose Chunk Annotator-Yamcha[2] uses TinySVM[3] for learning the classifiers. Yamcha is utilized to transform the input data into feature vectors usable by TinySVM [18,19]. Table 3 shows the Yamcha parameters. Accordingly, each classifier is unique in at least one of the following properties: window size, degree of the polynomial kernel, parsing direction as well as feature set. Consequently, this constructs 46 individual SVM classifier committees [17,20,21].

2.2 Generic Genetic Classifier-Ensemble Algorithm

The genetic algorithm (GA) was developed in the 1970s by Holland as an effective evolutionary optimization method [25]. In GA the two core elements are chromosome and fitness. Chromosome is used to encode representation of the optimal solution to the classifier-ensemble problem. Fitness is designed to measure the chromosome's performance.

Genetic Classifier-Ensemble-I. The basic idea behind the genetic classifier-ensemble-I is that different classes in each classifier differ with contributing degrees of prediction classes. In other words, each class in each classifier has been assigned a weight which corresponds with the contributing degree of prediction class. To use genetic algorithm, we first need to represent the problem domain as a chromosome. Here, we want to find an optimal set of weight for classifier ensemble scheme shown in Figure 1. Assume that there are totally N tags (classes) corresponding to the named entities considered in the BioNER task. Set the total number of available classifiers denoted by M. The optimal weight solution of the classifier ensemble scheme is encoded in the form of a weight chromosome,which has N*M genes. First N genes belong to the first classifier and the next N genes the second classifier and so on. The encoding of a chromosome is illustrated in Figure 1. Each value of gene in the chromosome is initialized to a small random number, said within the range[0,1]. Thus, we obtain a chromosome.

The second step is to define a fitness function for evaluating the chromosome's performance. This function must estimate the performance of a given classifier-ensemble

[2] http://cl.aist-nara.ac.jp/∼taku-ku/software/yamcha/

[3] http://chasen.org/∼taku/software/TinySVM/

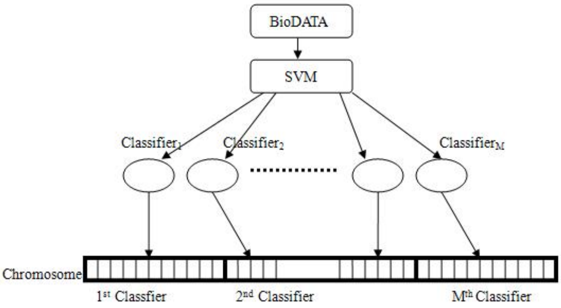


Fig. 1. Genetic Classifier-Ensemble-I

problem with weights. We define the fitness of a chromosome as the full object F-score provided by the weighted majority voting type decision combination rule [12,17,22]. In this rule, the class receiving the maximum combined score is selected as the joint decision. By the definition of the combined score of a particular class,

$$f(c_i) = \sum_{m=1}^{M} F_m \cdot w(m,i)$$

we obtain the fitness as follows:

$$f_n(c_l) = max(f(c_1), f(c_2), \cdots, f(c_n))$$

where M denotes the total number of classifiers and F_m denotes the full object F-score of *m*th classifier. *w(m,i)*is assigned to a weight value in the gene of *i*th class of *m*th classifier in the chromosome.

The third step is to choose the genetic operators-crossover and mutation. A crossover operator takes two parent chromosomes and creates two children with genetic material from both parents. In the proposed approach, either uniform or two point crossover method is randomly selected with equal probability. The selected operator is applied with a probability p_{cross} to generate two offspring. A mutation operator randomly selects a gene in offspring chromosomes with a probability p_{mut} and adds a small random number within the range[0,1] to each weight in the gene. In addition, we still need to specify the tournament size,elitism, population size and the number of generations. Tournament size is used in tournament selection during the reproduction. Elitism is applied at the end of each iteration where the best *elit_size%* of the original population are used to replace those in the offspring producing the lowest fitness.

Genetic Classifier-Ensemble-II. The basic principle behind the genetic classifier-ensemble-II is that different classifiers have different contributing degrees of prediction of classes. In other words, each classifier can be assigned a weight which corresponds with the contributing degree of prediction of class. Suppose each chromosome is encoded as a weight string having M genes, one for each classifier(see Figure 2). If the value of a gene is w_m, this means that the contributing degree of the *m*th classifier in this ensemble is w_m. Accordingly, the combined score of a given class can be redefined as:

$$f(c_i) = \sum_{m=1}^{M} F_m \cdot w_m$$

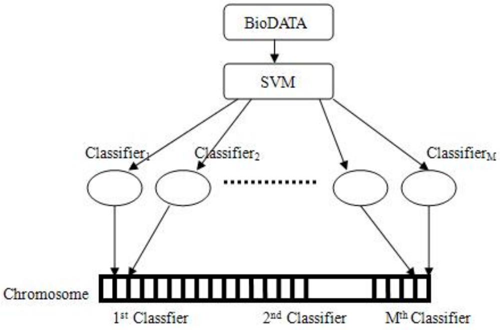


Fig. 2. Genetic Classifier-Ensemble-II

At the same time, all parameters of this algorithm described above including population size, the number of generations, crossover and mutation rate etc. are kept the same.

Genetic Classifier-Ensemble-III. Based on the above consideration in both subsections 2.2.1 and 2.2.2, not only contributing degrees of prediction classes among different classes in the same classifier are different, but also contributing degrees of prediction classes among different classifiers differ. Thus, the chromosome is made up of the chromosome in genetic classifier-ensemble-I and the chromosome in genetic classifier-ensemble-II, and has (N+1)*M genes (see Figure 3). Therefore, the combined score of a given class is determined as:

$$f(c_i) = \sum_{m=1}^{M} F_m \cdot w(m,i) \cdot w_m$$

Similarly, all the other parameters are kept the same.

After given the definition of chromosome and fitness as well as all parameters, the complete genetic classifier-ensemble algorithm can be described in the following steps:

1. Generate randomly an initial chromosome population of size **MAX_POPULATION**
2. For each chromosome in the population
   - 2.1 Apply weighted majority to all classifiers vector
   - 2.2 Compute full object **F-score** as fitness of the chromosome

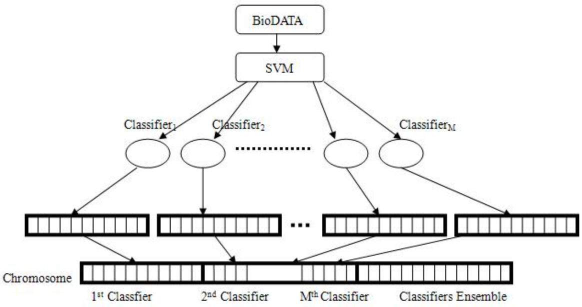


Fig. 3. Genetic Classifier-Ensemble-III

```
3. For generation_index in 1 ...MAX_GENERATION
   3.1 For chromosome_index in 1 ...MAX_POPULATION
       – Select two parents from the old population
       – Crossover the two parents to produce two offspring with probability $p_{cross}$
       – Mutate each gene of each offspring with probability $p_{mut}$
       – Apply weighted majority to each of the offspring
       – Compute full object F-score as fitness of each offspring
   3.2 Replace the worst ELIT_SIZE% of the offspring with the best chromosomes from the
       original population to form the new population
4. Select the best chromosome as the resultant ensemble
```

Figure 4 presents the flow of the proposed generic genetic classifier-ensemble algorithm.

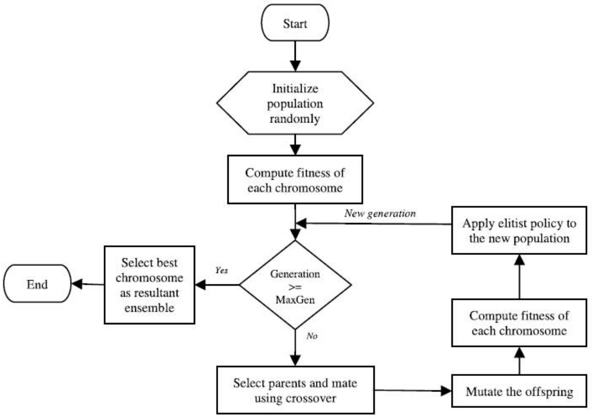

Start
Initialize population randomly
Compute fitness of each chromosome
New generation
Apply elitist policy to the new population
Generation >= MaxGen
Yes
No
Select best chromosome as resultant ensemble
End
Compute fitness of each chromosome
Select parents and mate using crossover
Mutate the offspring

Fig. 4. The flow of the proposed generic genetic classifier-ensemble algorithm

The overall system architecture is illustrated in Figure 5. The best-fitting solution of weighted classifier-ensemble is obtained by using the classifier outputs generated through three-fold cross-validation on the training data. In our proposed algorithm, the training data is initially partitioned into three parts. Each classifier is trained using two parts and then tested with the remaining part. This procedure is repeated three times and the whole set of training data is used for computing the best-fitting solution. Multi-class SVM is used for all individual classifier. The major differences among the individual classifiers are in their modeling parameter values and feature sets. Each classifier is different from the rest in at least one modeling parameter or the feature set. During testing, the outputs of the individual classifiers are combined by using the computed best-fitting solution of weight classifier-ensemble.

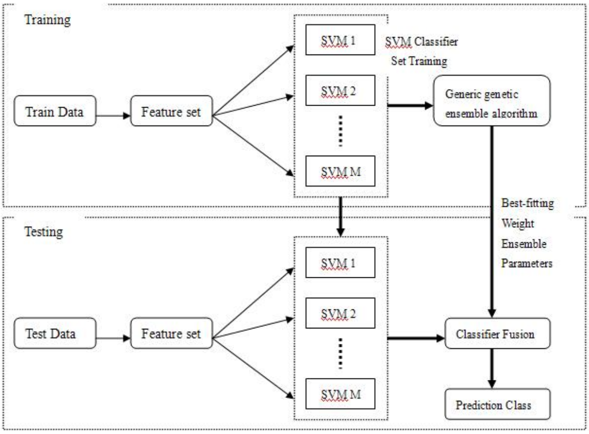


Fig. 5. Overall system architecture

3 Experiments and Results

To conduct the experiment, we use the latest GENIA[4] version 3.02 corpus provided by the shared task in COLING 2004 JNLPBA. The corpus includes the training dataset and the testing dataset. The training dataset consists of 2000 MEDLINE abstracts of the GENIA corpus with named entities in IOB2 format. The testing dataset consists of 404 abstracts. There are 18546 sentences and 492551 words in the training dataset and 3856 sentences and 101039 words in the testing dataset. Each word is tagged with "B-X", "I-X", or "O" to indicate that the word is at the "beginning"(B) or "inside"(I) of a named entity of type X, or "outside"(O) of a named entity. For BioNER task, the named entity types are DNA, RNA, cell_line, cell_type, and protein. Table 4 shows the number of 5 different biomedical named entities in this corpus. For each entity, two different tags(classes) result in 10 tags for the named entities and one additional tag for all non-named entities called class. Accordingly, this translate to a total of N=11 classes. Besides, we present M=46 single SVM base classifier committees on the basis

[4] http://www-tsujiii.is.s.u-tokyo.ac.jp/GENIA/

of different combination within feature set and Yamcha parameter. The experimental performance is evaluated by the standard measures, namely precision, recall and F-score which is the harmonic mean of precision and recall.

Table 4. Number of different biomedical named entities in GENIA 3.02 corpus

Types	Train data	Test data
DNA	9,534	1,056
RNA	951	118
Cell_line	3,830	500
Cell_type	6,718	1,921
Protein	30,269	5,067
Total	51,302	8,662

In the simulation experiments, The tournament size, crossover probability, mutation probability and elitism ratio are empirically computed as 40, 0.7, 0.02, and 20%, respectively. The population size of the generic genetic classifier-ensemble algorithm is set to 100. This means that one hundred different ensemble candidates evolve simultaneously. The algorithm is run for 10000 iterations. The weight classifier-ensemble corresponding to the chromosome with the highest fitness value in the last generation is selected as the optimal solution. We perform simulation experiments repeatedly by changing the weight values of these chromosomes and selected the weight genes of the chromosome providing the best performance of BioNER on the training data. In the testing, the test data is measured by using the optimal solution. This solution provides the best-fitting ensemble parameter with weights in the simulation experiments.

Table 5 shows the performance of the proposed three genetic classifier-ensemble scheme on precision, recall, and Fscore for BioNER. In this table, the genetic classifier-ensemble-III gets the better results compared with the genetic classifier-ensemble-I and genetic classifier-ensemble-II, where the performance of precision, recall and Fsore reach 75.65%, 78.52%, and 77.85% respectively.

It can be seen that in Table 6 we compare our best result with those of the recent work that employ support vector machines as classifier. The individual best SVM-classifier has the full feature set and optimal setting parameters[20,21]. Dimililer et al. used a vote-based classifier selection approach to construct a classifier ensemble and effective post-processing techniques for biomedical named entity recognition task[17,20,21]. Compared with the individual best SVM-classifier and SVM-classifier ensemble, our method outperforms them. It means that our generic genetic classifier-ensemble approach which searched the best-fitting ensemble parameter with weights can be powerful and efficient to combine orderly individual SVM base classifier with their strengths through giving the corresponding weights and to avoid individual classifier's weakness.

Table 7 shows that the best result of our experiment outperforms that of other individual classifier algorithms [26]. Their approaches include the Hidden Markov Model (HMM) [5], the Maximum Entropy Markov Model (MEMM) [4] and the Conditional Random Field (CRF) [3], which use deep knowledge resources with extra costs in

Table 5. The performances of different biomedical named entities on three genetic classifier-ensemble schemes

	Genetic Classifier-ensemble-I			**Genetic Classifier-ensemble-II**			**Genetic Classifier-ensemble-III**		
Types	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
DNA	73.54	74.25	72.92	70.68	70.98	70.76	74.65	75.59	75.21
RNA	74.33	75.85	75.22	71.15	70.25	70.46	75.88	76.79	76.42
Cell_line	72.50	71.56	72.12	68.25	67.20	67.82	74.60	73.82	74.36
Cell_type	73.15	72.87	72.04	69.62	72.58	70.37	74.85	75.39	75.06
Protein	83.36	76.58	79.65	80.56	71.25	75.86	84.58	80.06	83.57
Total	74.33	73.52	73.86	71.28	71.02	71.16	75.65	78.52	77.85

Table 6. The comparison with individual best SVM classifier and Vote-based SVM-classifier selection for bioNER task

Approaches	**Precision**	**Recall**	**F-score**
Single best SVM-classifier[20,21]	69.40	70.60	69.99
Vote-based SVM-classifier selection[20,21]	71.74	73.76	72.74
Genetic classifier-ensemble-III	75.65	78.52	77.85

Table 7. The comparison with other different individual classifier algorithms on bioNER task

Approaches	**Precision**	**Recall**	**F-score**
Zhou and Su[1]	69.42	75.99	72.55
Finkel et al.[2]	68.56	71.62	70.06
Settles[3]	69.30	70.30	69.80
Song et al.[4]	64.80	67.80	66.30
Zhao[5]	61.00	69.10	64.80
Genetic classifier-ensemble-III	75.65	78.52	77.85

pre-processing and post-processing. For instance, Zhou and Su [1] used name alias resolution, cascaded entity name resolution, abbreviation resolution and an open dictionary (around 700,000 entries). Finkel et al. used gazetteers and web-querying [2]. Settles used 17 lexicons that include Greek letters, amino acids, and so forth [3]. In contrast, our system did not include these similar processing.

4 Conclusion and Future Work

We proposed a generic genetic classifier-ensemble approach to recognizing the biomedical named entities. The contributions of this paper are that a novel genetic classifier-ensemble algorithm with weights is provided to deal with bioNER task and improve the BioNER performance compared with both of SVM-based classifiers as well as other individual machine learning algorithms. In the future, we will incorporate much more effective features and more classifiers using different machine learning algorithms in our ensemble approach, and include some post-processing techniques and comparison of computational cost.

References

1. Zhou, G., Su, J.: Exploring Deep Knowledge Resources in Biomedical Name Recognition. In: Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA 2004), pp. 70–75 (2004)
2. Finkel, J., Dingare, S., Nguyen, H., Nissim, M., Sinclair, G., Manning, C.: Exploiting Context for Biomedical Entity Recognition: From Syntax to the Web. In: Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications, JNLPBA 2004 (2004)
3. Settles, B.: Biomedical Named Entity Recognition Using Conditional Random Fields and Novel Feature Sets. In: Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA 2004), pp. 104–107 (2004)
4. Song, Y., Kim, E., Lee, G.-G., Yi, B.-K.: POSBIOTM-NER in the shared task of BioNLP/NLPBA 2004. In: Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications, JNLPBA 2004 (2004)
5. Zhao, S.: Name Entity Recognition in Biomedical Text using a HMM model. In: Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA 2004), pp. 84–87 (2004)
6. Zhang, Z., Yang, P.: An ensemble of classifiers with genetic algorithm based feature selection. IEEE Intelligent Informatics Bulletin 9, 18–24 (2008)
7. Yang, P., Zhang, Z., Zhou, B.B., Zomaya, A.Y.: Sample Subset Optimization for Classifying Imbalanced Biological Data. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part II. LNCS(LNAI), vol. 6635, pp. 333–344. Springer, Heidelberg (2011)
8. Yang, P., Yang, Y.-H., Zhou, B.B., Zomaya, A.Y.: A review of ensemble methods in bioinformatics. Current Bioinformatics 5, 296–308 (2010)
9. Yang, P., Ho, J.W.K., Zomaya, A.Y., Zhou, B.B.: A genetic ensemble approach for gene-gene interaction identification. BMC Bioinformatics 11, 524 (2010)
10. Kuncheva, L.I., Jain, L.C.: Designing classifier fusion systems by genetic algorithms. IEEE Transaction on Evolutionary Computation 4(4) (September 2000)
11. Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–5. Springer, Heidelberg (2000)
12. Ruta, D., Gabrys, B.: Application of the Evolutionary Algorithms for Classifier Selection in Multiple Classifier Systems with Majority Voting. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 399–408. Springer, Heidelberg (2001)
13. Larkey, L.S., Croft, W.B.: Combining classifier in text categorization. In: SIGIR 1996, pp. 289–297 (1996)
14. Patrick, J., Wang, Y.: Biomedical Named Entity Recognition System. In: Proceedings of the 10th Australasian Document Computing Symposium (2005)
15. Tsai, T.-H., Wu, C.-W., Hsu, W.-L.: Using Maximum Entropy to Extract Biomedical Named Entities without Dictionaries. In: JNLPBA 2006, pp. 268–273 (2006)
16. Chan, S.-K., Lam, W., Yu, X.: A Cascaded Approach to Biomedical Named Entity Recognition Using a Unified Model. In: The 7th IEEE International Conference on Data Mining, pp. 93–102

17. Dimililer, N., Varoğlu, E.: Recognizing Biomedical Named Entities Using SVMs: Improving Recognition Performance with a Minimal Set of Features. In: Bremer, E.G., Hakenberg, J., Han, E.-H(S.), Berrar, D., Dubitzky, W. (eds.) KDLL 2006. LNCS (LNBI), vol. 3886, pp. 53–67. Springer, Heidelberg (2006)
18. Kazamay, J.-I., Makinoz, T., Ohta, Y., Tsujiiy, J.-I.: Tuning Support Vector Machines for Biomedical Named Entity Recognition. In: ACL NLP, pp. 1–8 (2002)
19. Mitsumori, T., Fation, S., Murata, M., Doi, K., Doi, H.: Gene/protein name recognition based on support vector machine using dictionary as features. BMC Bioinformatics 6(suppl. 1) (2005)
20. Dimililer, N., Varoğlu, E., Altınçay, H.: Vote-Based Classifier Selection for Biomedical NER Using Genetic Algorithms. In: Martí, J., Benedí, J.M., Mendonça, A.M., Serrat, J. (eds.) IbPRIA 2007, Part II. LNCS, vol. 4478, pp. 202–209. Springer, Heidelberg (2007)
21. Dimililer, N., Varoglu, E., Altmcay, H.: Classifier subset selection for biomedical named entity recognition. Appl. Intell., 267–282 (2009)
22. Ruta, D., Gabrys, B.: Classifier selection for majority voting. Inf. Fusion 1, 63–81 (2005)
23. Yang, T., Kecman, V., Cao, L., Zhang, C., Huang, J.Z.: Margin-based ensemble classifier for protein fold recognition. Expert Syst. Appl. 38(10), 12348–12355 (2011)
24. Zhang, P., Zhu, X., Shi, Y., Wu, X.: An Aggregate Ensemble for Mining Concept Drifting Data Streams with Noise. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 1021–1029. Springer, Heidelberg (2009)
25. John, H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to biology, control and artificial intelligence. MIT Press (1998) ISBN 0-262-58111-6
26. Kim, J.-D., Ohta, T., Tsuruoka, Y., Tateisi, Y., Collier, N.: Introduction to the Bio-Entity Recognition Task at JNLPBA. In: Proceedings of the International Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA 2004), pp. 70–75 (2004)

Neighborhood Random Classification

Djamel Abdelkader Zighed[1], Diala Ezzeddine[2], and Fabien Rico[2]

[1] Institut des Sciences de l'Homme (ISH - USR 3385) Université de Lyon,
14, avenue Berthelot, 69007 Lyon
abdelkader.zighed@ish-lyon.cnrs.fr

[2] Laboratoire Eric, Université de Lyon,
5, avenue Pierre Mendès France, 69676 Bron Cedex, France
diala.ezzeddine@univ-lyon2.fr,
fabien.rico@univ-lyon1.fr

Abstract. Ensemble methods (EMs) have become increasingly popular in data mining because of their efficiency. These methods(EMs) generate a set of classifiers using one or several machine learning algorithms (MLAs) and aggregate them into a single classifier (Meta-Classifier, MC). Of the MLAs, k-Nearest Neighbors (kNN) is one of the most well-known used in the context of EMs. However, handling the parameter k can be difficult. This drawback is the same for all MLA that are instance based. Here, we propose an approach based on neighborhood graphs as an alternative. Thanks to these related graphs, like relative neighborhood graphs (RNGs) or Gabriel graphs (GGs), we provide a generalized approach with less arbitrary parameters. Neighborhood graphs have never been introduced into EM approaches before. The results of our algorithm : Neighborhood Random Classification are very promising as they are equal to the best EM approaches such as Random Forest or those based on SVMs. In this exploratory and experimental work, we provide the methodological approach and many comparative results.

Keywords: Ensemble methods, neighborhood graphs, relative neighborhood Graphs, Gabriel Graphs, k-Nearest Neighbors.

1 Introduction

Ensemble methods (EMs) have proved their efficiency in data mining, especially in supervised machine learning (ML). An EM generates a set of classifiers using one or several machine learning algorithms (MLA) and aggregates them into a single classifier (meta-classifier, MC) using, for example, a majority rule vote. Many papers [3,18,2,14] have shown that a set of classifiers produces a better prediction than the best among them, regardless of the MLA used. Theoretical and experimental results have encouraged the implementation of EM techniques in many fields of application such as physics [6], face recognition [17], ecology [12], recommender systems [9] and many others too numerous to mention here. The efficiency of EMs lies in the fact that aggregating different and independent classifiers reduces the bias and the variance of the MC [8,1,5,3], which are two key concepts for effective classifiers.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 98–108, 2012.

Instance based (IB) MLAs such as k-Nearest Neighbors (kNN) are very popular because of their straightforwardness. To implement them, it is simply necessary to define a dissimilarity measure on the set of observations and fix the value of k. Thus, using the kNN principle as an EM algorithm is immediate. However, handling the parameter k can be difficult for some users. To simplify this problem, we can use approaches based on neighborhood graphs as alternatives. For example, Relative Neighborhood Graphs (RNG) or Gabriel Graphs (GG) are "good" candidates. Like kNN, for an unlabeled observation, the classifier, based on neighborhood graphs, assigns a label according to the labels in the neighborhood. As an example, we can simply use the majority rule vote in the neighborhood of the unlabeled observation. While there have been many studies using kNN in the context of EM, we did not find any study that assesses the advantages of such neighborhood graphs, based more particularly on RNGs, in EM approaches. In this paper, we propose an EM approach based on neighborhood graphs. We provide comparisons with many EM approaches based on kSVM, Decision Tree (Random Forest), kNN etc. We carried out our experiments on an R platform.

This paper is organized as follows. In section 2, we introduce and recall certain notations and definitions. In section 3, we introduce the EMs based on neighborhoods. Besides the classic kNN neighborhood, we will present RNG and GG neighborhoods. Section 4 is devoted to evaluations and comparisons. Section 5 provides the main conclusions of this study.

2 Basic Concepts

2.1 Notations

Let Ω be a set of individuals represented by p attributes $X^j, j = 1, \ldots, p$ in a representing space $\Re$, and a membership class $Y \in \mathcal{K} = \{y_1, \ldots, y_K\}$. Let X be the function mapping an individual to its representation and Y the function mapping to the class :

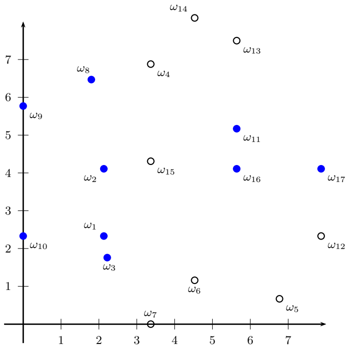
$$\begin{aligned} X \quad &: \quad \Omega \longrightarrow \Re \\ & \quad\quad \omega \longmapsto \left(X^j(\omega_i)\right)_{j=1,\ldots,p} \\ Y \quad &: \quad \Omega \longrightarrow \mathcal{K} \\ & \quad\quad \omega \longmapsto y \end{aligned}$$

Let us consider a training sample E_l of n individuals $(\omega_i)_{1\ldots n}$. For an individual ω_i, X_i denotes the vector $(X^j(\omega_i))_{j=1,\ldots,p}$ and Y_i its membership class $Y(\omega_i)$. Below, we will refer to an individual ω or to a point $X(\omega)$ with the same meaning, indistinguishably.

For the sake of illustration, we will use the toy example shown in Table 1. This is a two-class data set of 17 individuals mapped into a two-dimensional space $\Re = \mathbb{R}^2$.In the figure, the class $y_1 = 1$ is indicated by a bold dot and the class $y_2 = 2$ by an empty dot.

Table 1. Set of points in $\mathbb{R}^2$ with two classes

E_l	X^1	X^2	Y	E_l	X^1	X^2	Y
ω_1	2.13	2.33	2	ω_{10}	0	2.33	2
ω_2	2.13	4.11	2	ω_{11}	5.64	5.17	2
ω_3	2.22	1.76	2	ω_{12}	7.87	2.33	1
ω_4	3.37	6.88	1	ω_{13}	5.64	7.5	1
ω_5	6.77	0.67	1	ω_{14}	4.53	8.1	1
ω_6	4.53	1.16	1	ω_{15}	3.37	4.31	1
ω_7	3.37	0	1	ω_{16}	5.64	4.11	2
ω_8	1.8	6.47	2	ω_{17}	7.87	4.11	2
ω_9	0	5.77	2				



The goal of any machine learning algorithm is to produce a classifier capable of predicting, with high accuracy, the membership class $Y(\omega)$ for any individual ω whose attribute values $X(\omega)$ are known. Basically, the prediction is based on the knowledge we can obtain on the probability distribution:

$$P(Y/X) = (p(Y = y_k/X); k = 1, \ldots, K)\,.$$

Generally, a classifier ϕ helps, for all individuals ω, to estimate $\hat{P}$ which is the membership probability vector for all classes. Thanks to the learning sample E_l, the predicted membership class is $\hat{y_k}$ the most likely one, determined as follows:

$$\hat{y_k} \text{ is such that } \hat{p}(Y = \hat{y_k}/X) = \max_{k=1,\ldots,K} \hat{p}(Y = y_k/X)$$

By thresholding at the maximum value for this vector, the membership class can be represented by a zero vector except for the most likely class by a value of 1 at the corresponding rank: $\hat{P} = (0, \ldots, 0, 1, 0, \ldots, 0)$. If the classifier ϕ is considered as being reasonably reliable, then predicting of the membership class for an individual ω is $\phi(X(\omega)) \in \{y_1, \ldots, y_k, \ldots\}$.

2.2 Neighborhood Structure

There are many types of neighborhood that can be used to build a classifier. Among the most well known are:

- The well-known k-nearest neighbors;
- The ε-neighbors, which are defined by the subset of E_l that are in the ball of radius ε, centered on the individual, i.e. a point in Euclidean space;
- The neighborhood regions brought about by a decision tree where each leaf defines a subregion of the space. An individual that falls in a specific leaf has, as neighbors, those of the learning sample located in the same leaf.
- Parzens window neighbors;

- The neighbors in random spaces. For example, we can cite the weak models approach [7] where neighbors are obtained after a random projection along axes.
- The neighbors in the sense of a specific property. For example, GABRIEL Graph (GG) neighbors are given by the subset of individuals of the learning sample that fulfill a certain condition. Likewise, we can define the relative neighbors (RN), the minimum spanning tree's (MST) neighbors or the DELAUNAY's polyhedron neighbors and so forth [13];

2.3 Neighborhood Classifiers

The neighborhood classifiers depend on three components :

1. ***Neighborhood set*** $\mathcal{P}$: the set of all subsets of E_l . This is the set of all possible neighbors to which each individual will be connected.
2. ***The neighborhood function*** $\mathcal{V}$: this defines the way in which an individual is linked to an element in the neighborhood set:

$$\begin{aligned} \mathcal{V} : \Re &\longrightarrow \mathcal{P} \\ X &\longmapsto v = \mathcal{V}(X) \end{aligned}$$

   This function links any point X to a subset of E_l which contains its neighbors.
3. ***The decision rule*** **C**: this leads to probability distribution of the classes

$$\begin{aligned} C : \Re \times \mathcal{P} &\longrightarrow S_K \\ X, v &\longmapsto \Pi_v(X) = (p_1, p_2, \ldots, p_K) \end{aligned}$$

   where $S_K = \left\{(p_1, \ldots, p_K) \in [0,1]^K \text{ s.t. } \sum p_k = 1\right\}$

Hence, we can define a neighborhood classifier ϕ as based on a combination of the triplet $(\mathcal{P}, \mathcal{V}, C)$:

$$\phi(\omega) = \Pi_{\mathcal{V}(X(\omega))}(X(\omega))$$

2.4 Partition by Neighborhood Graphs

Here we focus on geometrical graphs, We thus build $\mathcal{P}$ using neighborhood graphs, such as VORONOI diagrams [13] or their dual (the DELAUNAY polyhedral), GABRIEL graphs [10], relative neighbors graphs [16] or the minimum spanning tree [11]. In such graphs, points are linked according to a specific property. Below we give the properties that define RNGs and GGs:

For a given distance measure d, a learning sample E_l and a set of individuals $\omega_1, \omega_2, \ldots$, any two points ω_i and ω_j are linked by the following rules :

- GABRIEL graph (GG) :

$$\omega_j \in \mathcal{V}_{GG}(\omega_i) \Longleftrightarrow \forall \omega \in E_l - \{\omega_i, \omega_j\} \quad d(\omega_i, \omega_j) \leq \sqrt{d^2(\omega_i, \omega) + d^2(\omega, \omega_j)};$$

– Relative neighbors graph (RNG) :

$$\omega_j \in \mathcal{V}_{RNG}(\omega_i) \Longleftrightarrow \forall \omega \in E_l - \{\omega_i, \omega_j\} \quad d(\omega_i, \omega_j) \leq \max\left(d(\omega_i, \omega), d(\omega, \omega_j)\right);$$

All these geometric structures induce a related neighborhood graph with a symmetric neighborhood relationship. Figures 1 and 2 show the neighbor structures of the relative neighbor graph and the Gabriel graph, using the dataset introduced above (cf 2.1).

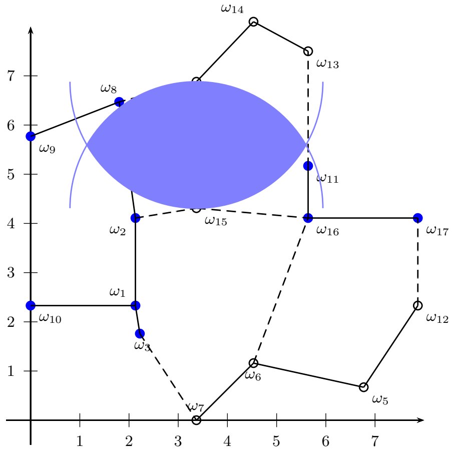


Fig. 1. Graph of relative neighbours

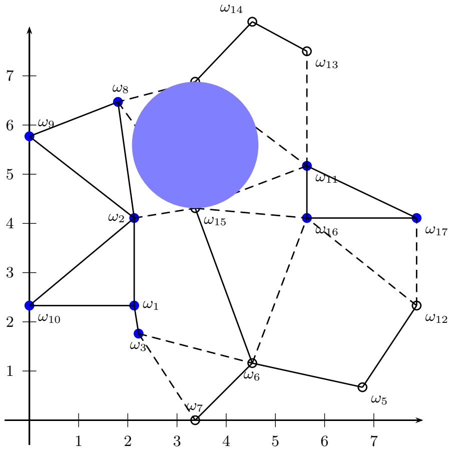


Fig. 2. Gabriel graph

3 Ensemble Method Classifier Based on Neighborhood

We call this framework "Random Neighborhood Classifier (RNC)". The principle of EMs is to generate M classifiers and then aggregate them into one (see 3). To do so, M randomized iterations are performed. At iteration m, RNC:

1. generates a new learning set E_l^m with a given size;
2. generates a new classifier $\phi^m = (\mathcal{P}^m, \mathcal{V}^m, C^m)$;
3. uses the generated classifier to determine the membership class of the unclassified individuals $\omega \in E_t$.

Following these steps, the RNC then aggregates the M predicted values related to an unclassified individual to determine its final membership class. The two key points in this procedure are the sampling procedure for generating the M classifiers and the procedure for combining the M predictions. Below, we provide some details of the two key points:

3.1 Sampling Procedures

From the training data set E_l which is an $n \times p$ table of values, we carry out M random samples. The sampling can be achieved in different ways:

- Sampling on rows with or without replacement;
- Sampling on columns;
- Building new columns by a linear combination of existing columns (oblique projection);
- Generating new individuals by a linear combination of columns;
- Randomly adding $x\%$ of rows and/or columns.

Each sample produced leads to a specific classifier.

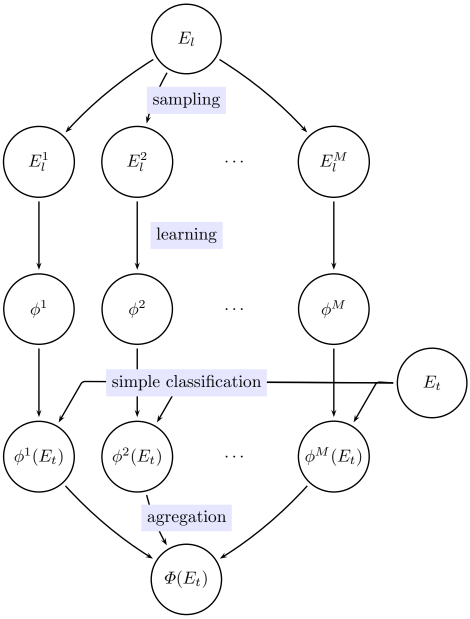


Fig. 3. EM procedure

3.2 Aggregating Function

Generally, the aggregating function is based on the majority rule vote. However, many other possibilities can be used [15]. Of these, we can cite:

- Vote of classifiers, which aggregate the responses of each classifier and normalize them. The majority rule vote is a particular example of this.
- Average vector where the score for each class is the mean of the answers for all the classifiers.
- Weighted version (majority or mean)
- Maximum Likelihood calculated as the product of the answers for all the classifiers, for each class. The winning class is the one that has the highest value.
- Naive Bayes [15].
- Decision Templates [15]. This method is based on the concept of a decision template, which is the average vector over the individuals of a test sample belonging to each class, and a decision profile, which is the set of responses of all classifiers. The membership class is determined according to the Euclidean distance between the decision profile and the decision template. The winning class is the one that minimizes this distance.
- Linear regression. In this method, we assume that the probability of a class is the linear combination of the probabilities of class for each classifier.

4 Evaluation

To assess the performance of RNC, we carried out many experiments on different data sets taken from the UCI Irvine repository. For this, we made a number of distinctions depending on the type of neighborhood used. As our work was motivated by the absence of studies on EMs based on geometrical graphs such as RNGs, we designed two separate experiments for RNC. One was based on RNGs and the other on kNN where k = 1, 2, 3. The comparison was also extended to random forests (RFs), K support vector machines (KSVMs), Adaboost, discriminant analysis (DA), logistic regression (RegLog) and C4.5. All experiments were carried out using R software.

4.1 Implementation of RNC

In our test, RNC use Relative Neighborhood classifiers. Each iteration uses the following scheme (see Figure 3) :

- **Sampling** : describing variables and individuals from learning sets are sampled (see the sampling procedure section 3 p.103).
  - For individuals we use a bootstrap for small data sets (less than 1000 individuals) and a proportion of 66% for the others.
  - For variables, we use a proportion of 50% of the variable for small dimensions (i.e. the dimension p is less than 20) and $\frac{10}{p}$ for the others.

  100 iterations were carried out, providing 100 classifiers.
- **Learning** : Mahalanobis distance between individuals is computed to construct the graph. The variance matrix needed to compute Mahalanobis distance is evaluated thanks to the learning set.
- **Simple classification** :
  - The neighborhood of a point $\omega \in E_t$ is computed using a Relative Neighbors Graph (RNG see section 2.3) :

    $$\mathcal{V}(\omega) = \mathcal{V}_{RNG}(\omega)$$

  - The decision rule is the proportion of each class in the Neighborhood :

    $$\begin{aligned} \phi(\omega) &= (p_1, \ldots, p_K) \\ \text{such that} \quad p_i &= \frac{\#\left(\{\omega' \in \mathcal{V}(\omega) s.t. Y(\omega') = i\}\right)}{\#\left(\mathcal{V}(\omega)\right)} \end{aligned}$$

    where $\#(\cdot)$ is the cardinal of a set.
- **Agregation** : the results are aggregated using Decision Templates (DT).

4.2 Other Methods

The implementation of RF is the `randomForest` library using 500 trees. We used the R `kernlab` library to apply the KSVM algorithm with classification type

C-svc. For DA and RegLog, we used the lda and glm functions of the R MASS library and for C4.5, the J48 function of the RWeka library using the control of the Weka learner.

For kNN, we simply replaced the neighborhood graph with the k-nearest neighbors in the RNC algorithm using the same distances and the same number of classifiers. The aggregation method is majority voting. Three values of k were tested : k = 1, 2 or 3.

4.3 The Test

We used 14 quantitative data sets. We ran the same protocol over all the methods mentioned above. For each experiment, we applied 10-Cross Validations to obtain an estimation of the error rates. For each dataset, we used the Wilcoxon test [4] to evaluate the results.

The results are shown in Table 2. For each dataset, we computed the average error rate, the rank of each method among the others and the p-values detected by the Wilcoxon test.

As can be seen in Table 2, RNCs based on RNGs performed well in comparison to kNN as well as in comparison to the other methods. Indeed, from the 14 data sets, RNC placed first once, second 6 times and third 4 times. RNCs were thus one of the 3 top methods in most cases.

We also computed the mean rank. This was done twice, using or not the classifiers adaBoost and Logistic Regression (which could not give an answer for more than 2 classes). The results are shown in Table 3 where the RNCs come out first. To see if the difference is significant, we applied the simple Friedman test [4], which showed a difference with a p-value of 6.849×10^{-6} and the post-hoc test (comparing each classifier by pair) gave the results shown in Table 4.

These results are very encouraging, because we believe that they can be improved by varying certain parameters such as:

- The choice of neighborhood structure, especially as we know that the neighborhood graphs are particularly sensitive to the dimension of the representation space.
- The type of base classifier. Should we use the closest connected homogeneous component? How can this notion of proximity be defined, precisely ? Should we take into consideration the size of the database or other characteristics of the neighborhood such as density, etc...?
- The selection methods to improve the quality of the data sets or the classifiers.

All these issues are currently being studied and should produce significant improvements for RNCs based on geometrical graphs.

Table 2. Comparison of RNC with several classification algorithms

	Glass			Image			Ionosphere			Iris		
	% Err.	Rank	Wilcox	% Err.	Rank	Wilcox	% Err.	Rank	Wilcox	% Err.	Rank	Wilcox
RNC	21.4	2		2.52	3		4	1		4.66	3	=
Random Forest	18.6	1	=	1.83	1	< 5%	6.29	4	=	5.33	4.5	=
KSVM	31.4	7	< 5%	5.57	7	< 1%	5.71	3	=	5.33	4.5	=
adaBoost	-	na		-	na		7.14	6	=	-	na	=
DA	38.5	8	< 5%	8.34	8	< 1%	12.3	10	< 1%	2	1	=
Log. Reg.	-	na		-	na		12	9	< 1%	-	na	=
C4.5	29.5	6	< 5%	3.17	5	=	8.86	7.5	< 5%	5.99	6.5	=
kNN1	25.7	3	=	2.17	2	=	6.58	5	< 5%	5.99	6.5	=
kNN2	27.6	5	=	3.34	6	< 5%	5.43	2	< 5%	8	8	=
kNN3	29.5	4	< 5%	3.13	4	< 5%	8.86	7.5	< 1%	4	2	=

	Letter (RvsB)			Musk			Diabete (Pima)			Ringnorm		
	% Err.	Rank	Wilcox	% Err.	Rank	Wilcox	% Err.	Rank	Wilcox	% Err.	Rank	Wilcox
RNC	1.05	2	=	3.02	3		24.9	6	=	2.26	2	
Random Forest	1.78	5	=	2	1.5	< 1%	23.	2	=	5.24	4	< 1%
Ksvm	1.38	4	=	3.46	5	=	24.1	5	=	1.56	1	=
adaBoostv	2.03	6	< 5%	2	1.5	< 1%	23.8	4	=	3.4	3	< 1%
DA	6.84	10	< 1%	5.58	10	< 1%	22.2	1	=	38.1	10	< 1%
Log. Reg.	6.25	9	< 1%	4.84	8	< 1%	23.7	3	=	34.8	9	< 1%
C4.5	5	8	< 1%	3.13	4	=	25.7	9	=	13.9	7	< 1%
kNN1	0.92	1	=	3.61	6	< 5%	25	7	=	6.37	5	< 1%
kNN2	3.48	7	< 5%	5.24	9	< 1%	30.1	10	=	24.2	8	< 1%
kNN3	1.32	3	=	4.49	7	< 1%	25.4	8	=	8.82	6	< 1%

	Sat			Sonar			Threenorm			Twonorm		
	% Err.	Rank	Wilcox	% Err.	Rank	Wilcox	% Err.	Rank	Wilcox	% Err.	Rank	Wilcox
RNC	10.8	4		12.5	2		13.3	2	=	2.64	3.5	
Random Forest	7.79	1	< 1%	14.5	4.5	=	13.6	3	=	3.4	7	< 5%
Ksvm	9.49	2	< 5%	17	7	=	12.4	1	=	2.69	5	=
adaBoostv	-	na		16	6	=	15.1	6	=	4.48	8	< 1%
DA	16.	8	< 1%	26	8	< 5%	16.6	7.5	< 1%	2.56	2	=
Log. Reg.	-	na		28	9	< 5%	16.6	7.5	< 1%	2.83	6	=
C4.5	13.7	7	< 1%	31	10	< 1%	28.5	10	< 1%	16.7	10	< 1%
kNN1	10.6	3	=	11	1	=	14.6	5	=	2.64	3.5	=
kNN2	12.3	6	< 5%	14.5	4.5	=	25.7	9	< 1%	10	9	< 5%
kNN3	11.9	5	< 5%	12.5	3	=	14.4	4	=	2.4	1	=

	Waveform			Wisc. Breast Cancer		
	% Err.	Rank	Wilcox	% Err.	Rank	Wilcox
RNC	14.8	4		2.65	2.5	
Rand. Forest	14.4	3	=	2.5	1	=
Ksvm	13.3	1	< 5%	4.12	9	< 5%
adaBoostv	-	na		3.09	5	=
DA	13.8	2	< 5%	3.82	8	=
Log. Reg.	-	na		3.24	6.5	=
C4.5	24.1	7	< 1%	4.41	10	< 5%
kNN1	16.9	6	< 1%	3.08	4	=
kNN2	29.6	8	< 1%	2.65	2.5	=
kNN3	15.9	5	< 1%	3.24	6.5	=

Table 3. Mean rank of the methods

	RNC	Random Forest	Ksvm	adaBoost	DA	Log. Reg.	C4.5	KNN1	KNN2	KNN3
All methods	2.88	3.19	4,04	5,06	6,58	7,56	7,46	4,15	7,04	4,58
Without adaBoost & LogReg	2.64	2.86	3.96		5.79		6.64	3.86	6.00	4.25

Table 4. p-value for the difference

	RNC	Random Forest	Ksvm	DA	C4.5	KNN1	KNN2	KNN3
RNC		1	0.84	0.015	0.0004	0.89	0.0067	0.66
Random Forest			0.93	0.032	0.0011	0.96	0.015	0.80
Ksvm				0.50	0.072	1	0.35	1
DA					0.98	0.42	1	0.71
C4.5						0.052	1	0.16
KNN1							0.28	1
KNN2								0.55
KNN3								

4.4 Computational Analysis

The theoretical complexity for graph computation using n individuals represented by p variables is $O(n^3 + n^2p^2)$. Indeed, for each pair of individuals ω_1, ω_2, it is necessary to test the RNG condition :

$$\forall \omega \in E_l - \{\omega_1, \omega_2\} \quad d(\omega_1, \omega_2) \leq \max\left(d(\omega_1, \omega), d(\omega, \omega_2)\right); \tag{1}$$

So, this means computing the distance $d(\omega_1, \omega_2)$ for each possible pair ($O(p^2)$ each for Mahalanobis) and compare the distance $d(\omega_1, \omega 2)$ with all distance $d(\omega_1, \omega)$ dans $d(\omega_2, \omega)$ for each individuals ω.

But optimization can be carried out :

- Distance can be computed using matrix representation and a powerful linear algrebra library (BLAS).
- Using the RNG condition, it is only necessary to test 1 for all ω such that $d(\omega_1, \omega) \leq d(\omega_1, \omega_2)$.
- For a given individual ω_1, sorting all distance $d(\omega_1, \omega)$ by increasing order. Then, we test the condition 1 following this order. If the distance $d(\omega_1, \omega_2)$ is large compared to the others, we reject the edge between ω_1 and ω_2 faster.

Practically, computing the results using RNG graphs is several times slower than using k-NN. For example with k=3, and the data set Twonorm (almost 2000 individuals), it takes 40s for k-NN method and 1min44s for the RNG method to compute N=100 classifiers. These tests use the BLAS library `atlas`, on a Intel$^{\text{TM}}$ Core$^{\text{TM}}$ i5 2.60GHz computer with 4G memory.

5 Conclusion and Further Work

Here we have provided a new approach for using neighborhood structures in ensemble methods. The results obtained show that they are challenging the most powerful techniques such as random forests and kSVM. Methods based on geometrical neighborhood graphs outperform the classic methods such as *kNN*. There are many possibilities for improving RNC based on RNG. A library containing all the functionalities that have been achieved is available by emailing the authors.

References

1. Breiman, L.: Bias, variance, and arcing classifiers. Statistics (1996)
2. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
3. Brown, G., Wyatt, J., Harris, R., Yao, X.: Diversity creation methods: a survey and categorisation. Information Fusion 6(1), 5–20 (2005)
4. Demsar, J.: Statistical comparisons of classifiers over multiple data sets (2006)
5. Domingos, P.: A unified bias-variance decomposition and its applications. In: ICML, pp. 231–238. Citeseer (2000)
6. Ham, J.S., Chen, Y., Crawford, M.M., Ghosh, J.: Investigation of the random forest framework for classification of hyperspectral data. IEEE Transactions on Geoscience and Remote Sensing 43(3) (2005)
7. Ho, T., Kleinberg, E.: Building projectable classifiers of arbitrary complexity. In: International Conference on Pattern Recognition, vol. 13, pp. 880–885 (1996)
8. Kohavi, R., Wolpert, D.: Bias plus variance decomposition for zero-one loss functions. In: Machine Learning-International Workshop, pp. 275–283. Citeseer (1996)
9. O'Mahony, M.P., Cunningham, P., Smyth, B.: An Assessment of Machine Learning Techniques for Review Recommendation. In: Coyle, L., Freyne, J. (eds.) AICS 2009. LNCS, vol. 6206, pp. 241–250. Springer, Heidelberg (2010), http://portal.acm.org/citation.cfm?id=1939047.1939075
10. Park, J.C., Shin, H., Choi, B.K.: Elliptic Gabriel graph for finding neighbors in a point set and its application to normal vector estimation. Computer-Aided Design 38(6), 619–626 (2006)
11. Planeta, D.S.: Linear time algorithms based on multilevel prefix tree for finding shortest path with positive weights and minimum spanning tree in a networks. CoRR abs/0708.3408 (2007)
12. Prasad, A.M., Iverson, L.R., Liaw, A.: Newer classification and regression tree techniques: bagging and random forests for ecological prediction. Ecosystems 9(2), 181–199 (2006)
13. Preparata, F.P., Shamos, M.I.: Computational geometry: an introduction. Springer (1985)
14. Schapire, R.: The boosting approach to machine learning: An overview. Lecture Note in Statistics, pp. 149–172. Springer (2003)
15. Shipp, C.A., Kuncheva, L.I.: Relationships between combination methods and measures of diversity in combining classifiers. Information Fusion 3(2), 135–148 (2002)
16. Toussaint, G.T.: The relative neighbourhood graph of a finite planar set. Pattern Recognition 12(4), 261–268 (1980)
17. Wang, X., Tang, X.: Random sampling lda for face recognition, pp. 259–267 (2004), http://portal.acm.org/citation.cfm?id=1896300.1896337
18. Zhou, Z.H., Wu, J., Tang, W.: Ensembling neural networks: Many could be better than all* 1. Artificial Intelligence 137(1-2), 239–263 (2002)

SRF: A Framework for the Study of Classifier Behavior under Training Set Mislabeling Noise

Katsiaryna Mirylenka[1], George Giannakopoulos[2], and Themis Palpanas[1]

[1] University of Trento, Italy
[2] Institute of Informatics and Telecommunications of NCSR Demokritos, Greece
kmirylenka@disi.unitn.eu, ggianna@iit.demokritos.gr, themis@disi.unitn.eu

Abstract. Machine learning algorithms perform differently in settings with varying levels of training set mislabeling noise. Therefore, the choice of a good algorithm for a particular learning problem is crucial. In this paper, we introduce the "Sigmoid Rule" Framework focusing on the description of classifier behavior in noisy settings. The framework uses an existing model of the expected performance of learning algorithms as a sigmoid function of the signal-to-noise ratio in the training instances. We study the parameters of the above sigmoid function using five different classifiers, namely, Naive Bayes, kNN, SVM, a decision tree classifier, and a rule-based classifier. Our study leads to the definition of intuitive criteria based on the sigmoid parameters that can be used to compare the behavior of learning algorithms in the presence of varying levels of noise. Furthermore, we show that there exists a connection between these parameters and the characteristics of the underlying dataset, hinting at how the inherent properties of a dataset affect learning. The framework is applicable to concept drift scenaria, including modeling user behavior over time, and mining of noisy data series, as in sensor networks.

Keywords: classification, classifier evaluation, handling noise, concept drift.

1 Introduction

Transforming vast amounts of collected — possibly noisy — data into useful information, through such processes as clustering and classification, is a really interesting research topic. The machine learning and data mining communities have extensively studied the behavior of classifiers — which is the focus of this work — in different settings (e.g., [13,20,8,9]), however the effect of noise on the classification task is still an interesting and open problem. The importance of studying noisy data settings is augmented by the fact that noise is very common in a variety of large scale data sources, such as sensor networks and the Web. Thus, there rises a need for a unified framework studying the behavior of learning algorithms in the presence of noise, regardless of the specifics of each algorithm.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 109–121, 2012.

In this work, we study the effect of training set mislabeling noise[1] on a classification task. This type of noise is common in cases of *concept drift*, where a target concept shifts over time, rendering previous training instances obsolete. Essentially, in the case of concept drift, feature noise causes the labels of previous training instances to be obsolete and, thus, equivalent to mislabeling noise. Drifting concepts appear in a variety of settings in the real world, such as the state of a free market or the traits of the most viewed movie. Giannakopoulos and Palpanas [10] have shown that the performance[2] of a classifier in the presence of noise can be effectively approximated by a *sigmoid* function, which relates the signal-to-noise ratio in the training set to the expected performance of the classifier. We term this approach the "Sigmoid Rule".

In our work, we examine how much added benefit we can get out of the sigmoid rule model, by studying and analyzing the parameters of the sigmoid in order to detect the influence of each parameter on the learner's behavior. Based on the most prominent parameters, we define the dimensions characterizing the algorithm behavior, which can be used to construct criteria for the comparison of different learning algorithms. We term this set of dimensions the *"Sigmoid Rule" Framework (SRF)*. We also study, using SRF, how dataset attributes (i.e., the number of classes, features and instances and the fractal dimensionality [6]) correlate to the expected performance of classifiers in varying noise settings.

In summary, we make the following contributions. We define a set of intuitive criteria based on the SRF that can be used to compare the behavior of learning algorithms in the presence of noise. This set of criteria provides both quantitative and qualitative support for learner selection in different settings. We demonstrate that there exists a connection between the SRF dimensions and the characteristics of the underlying dataset, using both a correlation study and regression modeling. In both cases we discovered statistically significant relations between SRF dimensions and dataset characteristics. Our results are based on an extensive experimental evaluation, using 10 synthetic and 14 real datasets originating from diverse domains. The heterogeneity of the dataset collection validates the general applicability of the SRF.

2 Background and Related Work

Given the variety of existing learning algorithms, researchers are often interested in obtaining the best algorithm for their particular tasks. This algorithm-selection is considered part of the meta-learning domain [11]. According to the No-Free-Lunch theorems (NFL) described in [22] and proven in [23], [21], there is no overall best classification algorithm. Nevertheless, NFL theorems, which compare the learning algorithms over diverse datasets, do not limit us when we focus on a particular dataset. As mentioned in [1], the results of NFL theorems

[1] For the rest of this paper we will use the term *noise* to refer to this type of noise, unless otherwise indicated.

[2] In this paper, by *performance* of an algorithm, we mean classification accuracy.

hint at comparing different classification algorithms on the basis of dataset characteristics. Concerning the measures of performance that help distinguish among learners, in [1] the authors compared algorithms on a large number of datasets (100), using measures of performance that take into consideration the distribution of the classes within the dataset, thus using the characteristics of datasets. The Area Under the receiver operating Curve (AUC) is another measure used to assess machine learning algorithms and to divide them into groups of classifiers which have statistically significant difference in performance [2]. In all the above studies, the analysis of performance has been applied on datasets without noise, while we study the behavior of classification algorithms in noisy settings. Our present study is based on the work of G. Giannakopoulos and T. Palpanas [10] on concept drift, which illustrated that a sigmoid function can efficiently describe performance in the presence of varying levels of training set mislabeling noise. In this work, we analytically study the sigmoid function to determine a set of parameters that can be used to support learner selection in different noisy classification settings.

The behavior of machine learning classifiers in the presence of noise was also considered in [14]. The artificial datasets used for classification were created on the basis of predefined linear and nonlinear regression models, and noise was injected in the features, instead of the class labels as in our case. Noisy models of non-markovian processes using reinforcement learning algorithms and Temporal Difference methods are analyzed in [18]. In [4], the authors examine multiple-instance induction of rules for different noise models. There are also theoretical studies on regression algorithms for noisy data [19] and works on denoising, like [17], where a wavelet-based noise removal technique was shown to increase the efficiency of four considered machine learners. In both noise-related studies [19], [17] attribute noise was considered. However, we study class-related noise and do not consider specific noise models, which is a different problem. Class-related noise is mostly related to concept drift, as was also discussed in the introduction. In an early influential work, the problem of *concept attainment* in the presence of noise was indicated and studied in the STAGGER system [16]. To the best of our knowledge, there has been no work related to the selection of a classifier in a concept drift setting, based on the level of noise and other qualitative criteria, which will be reported below.

3 The Sigmoid Rule Framework

In order to describe the performance of a classifier, the “sigmoid rule” of [10] considers a function which relates signal-to-noise ratio of the training set to the expected performance. This function is called the *characteristic transfer function* (CTF) of a learning algorithm. In this work we will call it also the *sigmoid function* of an algorithm. The function is of the form

$$f(Z) = m + (M - m)\frac{1}{1 + b \cdot \exp(-c(Z - d))}$$

where $m \leq M$; $b, c > 0$; $Z = log(1+S) - log(1+N)$; S is the amount of "signal" or true data, while N is the amount of "noisy" or distorted data; hence, Z is the signal-to-noise ratio. As was shown in [10] the sigmoid function effectively approximates the performance of a classifier in noisy settings.

The behavior of different machine learning algorithms in the presence of noise can be compared on several *axes of comparison*, based on the sigmoid function parameters. Related to *performance* we can use (a) the minimal performance m; (b) the maximal performance M; (c) the width of the performance range $r_{alg} = M - m$, that defines the width of the interval in which the algorithm performance varies. Related to the *sensitivity* of performance to the change of the signal-to-noise ratio we want to know (a) within which range of noise levels there is a significant change in performance when changing the noise; (b) how we can tell apart algorithms that improve their performance even when the signal-to-noise levels are low over those which only improve in high ranges of signal-to-noise ratio; (c) how we can measure the stability of performance of an algorithm against varying noise; (d) at what noise level an algorithm reaches its average performance. To address these requirements we perform an analytic study of the sigmoid CTF of an algorithm. This analysis helps devise measurable dimensions that can answer our questions.

The domain of the sigmoid is in the general case $Z \in (-\infty, +\infty)$. The range of values is (m, M). Based on the first three derivatives, we determine the point $Z_{inf} = d + \frac{1}{c}\log b$, which is *the point of inflection* (curvature sign change point). In the case of the sigmoid function, this point is also the centre of symmetry. Furthermore, Z_{inf} indicates the shift of the sigmoid with respect to the origin of the axes. The zeros of the third order derivative are $Z_{1,2}^{(3)} = d - \frac{1}{c}\log\frac{2\pm\sqrt{3}}{b}$, which can be used to estimate the slope of the sigmoid curve. Figure 1 illustrates the sigmoid curve and its points of interest.

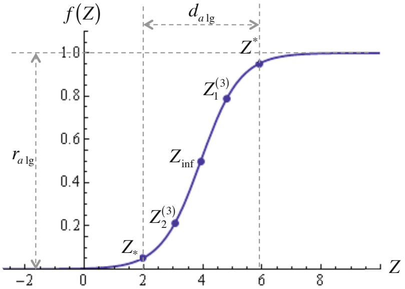


Fig. 1. Sigmoid function and points of interest

In the following section, we formulate and discuss dimensions that describe the behavior of algorithms, based on our axes of comparison.

3.1 Sigmoid Rule Framework (SRF) Dimensions

We define several SRF dimensions based on the sigmoid properties, in addition to m, M, r_{alg} defined in Section 3. We define as *active noise range* a range $[Z_*, Z^*]$ where the change of noise induces a measurable change in the performance. To calculate $[Z_*, Z^*]$, let us assume that there is a good-enough performance for a given task, approaching M for a given algorithm. We know that $f(Z) \in (m, M)$ and we say that the performance is good enough if $f(Z) = M - (M - m) * p, p = 0.05$[3]. We define the size of the signal-to-noise interval in which $f(Z) \in [m + (M - m) * p, M - (M - m) * p]$ to be the *learning improvement* of the algorithm. Then, using the inverse function $f^{-1}(y)$ we calculate the points Z^* (corr. Z_*) which is the bottom (corr. top) point in Figure 1 for a given p. We term the distance $d_{alg} = Z^* - Z_*$ as the *width of the active area* of the machine learning classifier (see Figure 1). Then, $\frac{r_{alg}}{d_{alg}}$ describes the learning performance improvement over signal-to-noise ratio change; we term this measure the *slope indicator*, as it is indicative of the slope of the CTF.

In the following paragraphs we describe how the analysis of the CTF allows to compare learning algorithm performance in the presence of noise.

3.2 Comparing Algorithms

Given the performance dimensions described above, we can compare algorithms as follows. For *performance* we can use: m, M, r_{alg}. Algorithms not affected by the presence or absence of noise will have a minimal r_{alg} value. In a setting with a randomly changing level of noise this parameter is related to the possible variance in performance. Related to the *sensitivity* of performance to the change of the signal-to-noise ratio we can use: (a) the *active noise range* $[Z_*, Z^*]$. The width of the active area of the algorithm $d_{alg} = Z^* - Z_*$, which is related to the speed of changing performance for a given r_{alg} in the domain of noise. A high d_{alg} value indicates that an algorithm varies its performance in a broad range of signal-to-noise ratios, implying less stability of performance in an environment with heavily varying degrees of noise. We say that the algorithm *operates* when the level of noise in the data is within the active noise range of the algorithm; (b) the lower bound Z_* of the active noise range, which suggests which algorithm operates earlier in noisy environment and which can reach its maximal performance fast; (c) the point of inflection Z_{inf}, that shows the signal-to-noise ratio for which an algorithm gives the average performance. Z_{inf} can be used to choose the algorithm that reaches its average performance under more noise.

A parameter related to both *performance* and *sensitivity* is the slope indicator $\frac{r_{alg}}{d_{alg}}$. It can be used to determine whether reducing the noise in a dataset is expected to have a significant impact on the performance. An algorithm with a high value of $\frac{r_{alg}}{d_{alg}}$, implies that reducing noise would be very beneficial. Furthermore, using the same dimension one can choose more stable algorithms, when the

[3] The value 0.05 can be any value close to 0, describing a normalized measure of distance from optimal performance.

variance of noise is known. In this case, one may choose the algorithm with the lowest value of $\frac{r_{alg}}{d_{alg}}$, in order to limit the corresponding variance in performance. Based on the above discussion, we consider the algorithms with higher maximal performance M, larger width of performance range r_{alg}, higher slope indicator $\frac{r_{alg}}{d_{alg}}$ and shorter width of the active area of the algorithm d_{alg} to behave better: we expect to get high performance from an algorithm if the level of noise in the dataset is very low, and low performance if the level of noise in the dataset is very high. Decision makers can easily formulate different criteria, based on the proposed dimensions and particular settings.

4 Experimental Evaluation

In the following paragraphs, we describe the experimental setup, the datasets and the results of our experiments.

In our study, we used the following machine learning algorithms, implemented in Weka 3.6.3 [12]: (a) IBk — K-nearest neighbor classifier; (b) Naive Bayes classifier; (c) SMO — support vector classifier (cf. [15]); (d) NbTree — a decision tree with naive Bayes classifiers at the leaves; (e) JRip — a RIPPER [5] rule learner implementation. We have chosen representative algorithms from different families of classification approaches, covering very popular classification schemes [24].

We used a total of 24 datasets for our experiments.[4] Fourteen of them are real, and ten are synthetic. All the datasets were divided into groups according to the number of classes, attributes (features) and instances in the dataset as is shown on Figure 2. There are 12 possible groups that include all combinations of the parameters. Two datasets from each group were employed for the experiments.

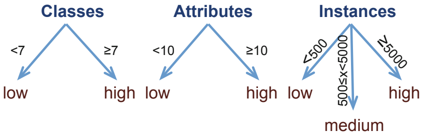


Fig. 2. Datasets grouping labels

We created artificial datasets in the cases were real datasets with a certain set of characteristics were not available. We produced datasets with known intrinsic dimensionality. The distribution of dataset characteristics is illustrated in Figure 3. The traits of the datasets illustrated are the number of classes, the number of attributes, the number of instances and the estimated intrinsic (fractal) dimension.

The ten artificial datasets we used were built using the following procedure. Having randomly sampled the number of classes, features and instances, we

[4] Most of the real datasets come from the UCI Machine learning repository [7], and one from [10]. For a detailed list with references check the following anonymous online resource: `http://tinyurl.com/3g4fmsf`.

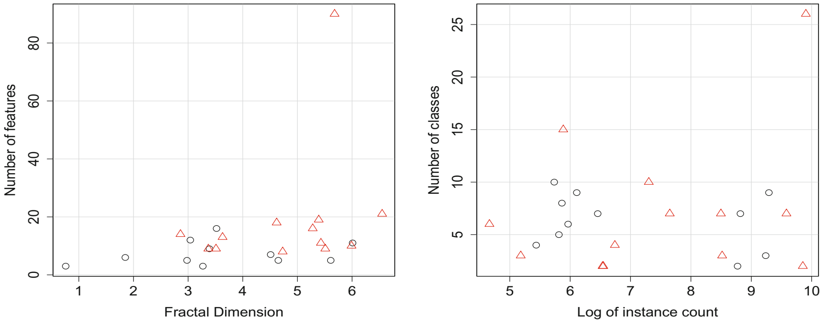


Fig. 3. Distribution of real (triangles) and artificial (circles) dataset characteristics

sample the parameters of each feature distribution. We assume that the features follow the Gaussian distribution with mean value (μ) from the interval $[-100, 100]$ and standard deviation(σ) from the interval $[0.1, 30]$. The μ and σ intervals allow overlapping features across classes.

Noise was induced as follows. We created stratified training sets, equally sized to the stratified test sets. To induce noise, we created noisy versions of the training sets by mislabeling instances. Using different levels l_n of noise, $l_n = 0, 0.05, ..., 0.95$[5], a training set with l_n noise is a set where there is a l_n probability that a training instance will be assigned a different label than their true one. Hence, we obtained 20 dataset versions with varying noise levels.

4.1 Using SRF

We performed experiments of "noisy" classification using the generated datasets, performing 10-fold cross validation per algorithm, and calculated the average performance for varying noise levels. Given the 20 levels of signal-to-noise ratio and the corresponding algorithm performance, (i.e., classification accuracy) we estimated the parameters of the sigmoid. The search in the parameter space is performed by a genetic algorithm, estimating an approximate good set of parameters as was proposed in [10]. The quality of estimation is checked using the Kolmogorov-Smirnov test. The results obtained are statistically significant.

A sample of true and sigmoid-estimated performance graphs for varying levels of noise can be seen in Figure 4. In our experiments, the parameters of the sigmoid were estimated offline, but SRF can be applied in an online scenario, as well, using a training period.

Figure 5 illustrates the means of the SRF parameters per algorithm, over all 24 datasets. As an example of interpretation of the figure using SRF, the plots

[5] We note that high levels of noise such as 95% are often observed in the presence of *concept drift*, e.g., when learning computer-user browsing habits in a network environment with a single IP, and several different users sharing it.

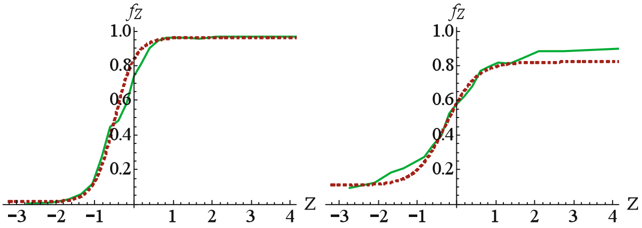


Fig. 4. Sigmoid CTF of SMO (left) and IBk (right) for "Wine" dataset. Green solid line: True Measurements, Dashed red line: estimated sigmoid.

indicate that (for the studied range of datasets) SMO is expected to improve its performance faster than all other algorithms, when the signal-to-noise ratio increases. This conclusion is based on the slope indicator ($\frac{r_{alg}}{d_{alg}}$) values. Also, IBk has a smaller potential for improvement of performance (but also smaller potential for loss) than SMO when noise levels change, given that the width of the performance range r_{alg} is higher for SMO. This difference can also be seen in Figure 4, where the distance between minimum and maximum performance values is bigger for the SMO case (see Figure 4(left)).

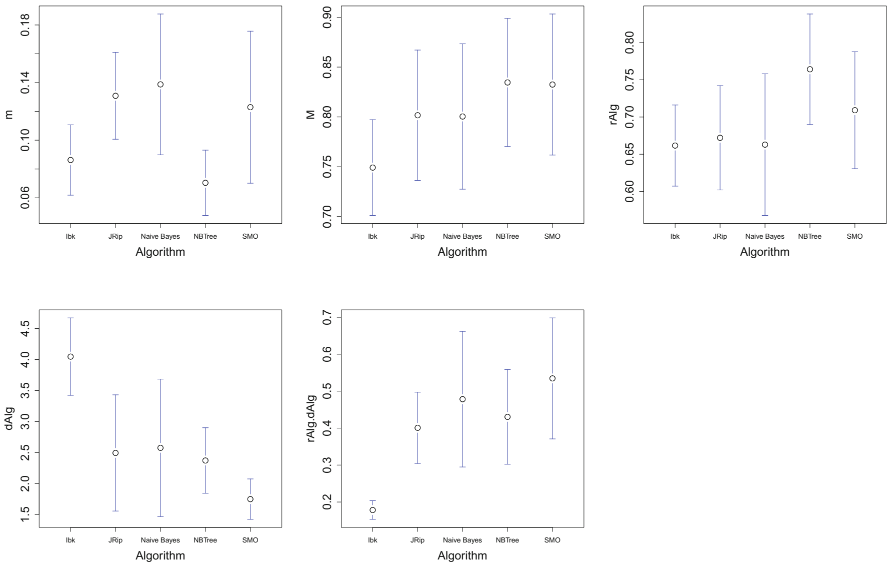


Fig. 5. SRF parameters per algorithm. X axis labels (left-to-right): IBk, JRip, NB, NBTree, SMO.

We stress that parameter estimation does not require previous knowledge of the noise levels, but it is dataset dependent. In the special case of a classifier selection process, having an estimate of the noise level in the dataset helps to reach a decision through the use of SRF.

4.2 Statistical Analysis

We now study the connection between the dataset characteristics and the sigmoid parameters (using the same 24 datasets), irrespective of the choice of the algorithm. We consider the results obtained from all the algorithms as different samples of SRF parameters for a particular dataset. We use regression analysis to observe the cumulative effect of the dataset characteristics on a single parameter, and we use correlation analysis to detect any connection between each (dataset characteristic, sigmoid parameter) pair. We examine the connections between dataset characteristics and the sigmoid parameters both individually, and all together, in order to draw the complete picture.

Regression Analysis. We wanted to examine how the number of classes (x_1), number of features (x_2), number of instances (x_3), and intrinsic dimensionality[6] (as fractal correlation dimension [3]) (x_4) of a dataset influence the CTF parameters.

We applied a leave-one-out process, where one dataset is left out from training and used for testing on every run. We used in turn m, M, r_{alg}, d_{alg}, and $\frac{r_{alg}}{d_{alg}}$ as dependent variables. The results of model fitting and prediction of SRF dimensions are reported in Table 1, where average errors between observed and predicted SRF dimensions are shown. For each SRF dimensions chosen, we have observed 5 values (since 5 machine learning algorithms were used), and having estimated them for 24 datasets, we end up with 120 predictions for a single SRF dimension. We calculated four types of errors: (1) MSE — mean square error; (2) MAE — mean absolute error; (3) RMSE — relative mean square error and (4) RMAE — relative mean absolute error. The last column of Table 1 shows the average of the adjusted R^2 statistic for models that where estimated for all the SRF dimensions (average on the 24 datasets). Figure 6 illustrates how our models fit the test data, showing that in most cases the true values of the sigmoid parameters for each dataset (illustrated by circles that correspond to 5 algorithms for each test dataset i, $i = 1, 2, ..., 24$) are within the 95% confidence level zone around the estimated values. This finding further supports the connection between the dataset parameters and SRF dimensions. According to the results, the chosen parameters of the datasets can be used to predict the parameters of the sigmoid of the algorithms.

Correlation Analysis. We used three different correlation coefficients — Pearson correlation for linear correlation, Spearman's rho and Kendall's tau for

[6] The authors would like to thank Christos Faloutsos for kindly providing the code for the fractal dimensionality estimation.

Table 1. Prediction error of linear regression models

Parameters	**Error measures**				**average(R_a^2)**
	MSE	MAE	RMSE	RMAE	
m	0.11	0.09	148.92	29.17	0.54
M	0.35	0.30	0.51	0.41	0.88
r_{alg}	0.32	0.27	0.71	0.46	0.85
d_{alg}	1.98	1.41	0.97	0.68	0.67
$\frac{r_{alg}}{d_{alg}}$	0.37	0.27	4.83	1.46	0.55

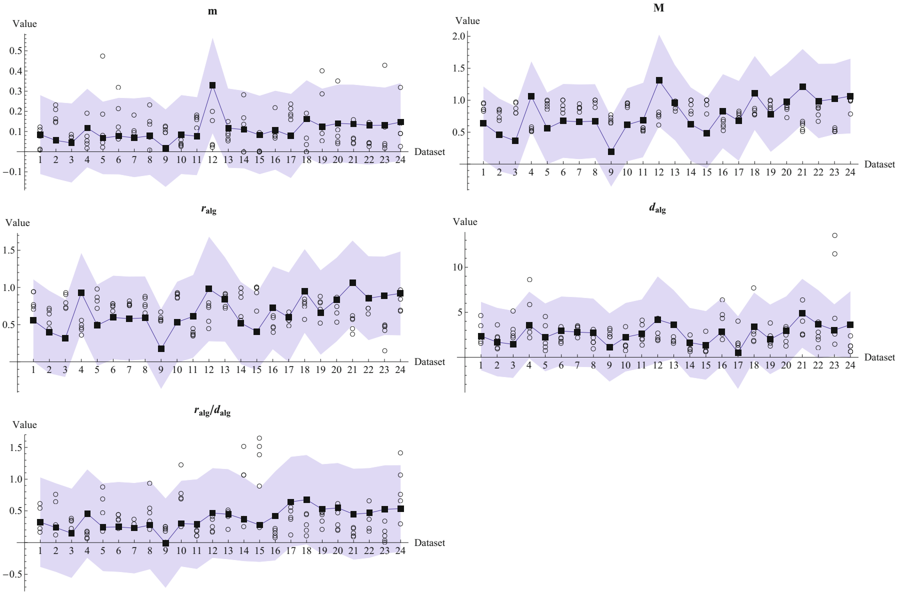


Fig. 6. Real and estimated values of the sigmoid parameters. Real values: Black rectangles, Estimated values: circles, Gray zone: 95% prediction conf. interval.

monotonic correlation — to analyze the connection between the parameters of the datasets and the CTF parameters (cf. Table 2). We qualitatively interpret the strength of the correlation as follows: $[0.0; 0.1)$ →No Correlation, $[0.1; 0.3)$ →Low Correlation, $[0.3; 0.5)$ →Medium Correlation, $[0.5; 1]$ →Strong Correlation.

Summarizing the results from all the correlation coefficients (refer to Table 2), some interesting conclusions can be drawn. First, the number of classes (x_1) is inversely correlated to $\frac{r_{alg}}{d_{alg}}$, r_{alg} and M. Thus, the higher the number of classes is, the lower the sensitivity to noise variation (check on $\frac{r_{alg}}{d_{alg}}$); the lower the number of classes, the higher the impact of reducing noise on performance (check r_{alg} and M). These conclusions are also supported by the direct correlation between the number of classes and the width of the active area of the algorithm d_{alg}. We also note the complete lack of significant correlation between the minimum performance m and all of the SRF dimensions: given enough noise an algorithm always

Table 2. Correlation between dataset parameters and SRF parameters. Colored cells: statistically significant correlation ($p - value < 0.05$: **underlined bold**, $p - value < 0.1$: ***italics-bold***). Green (dark) : medium correlation, gray (light) : low correlation.

		Pearson's corr.				Spearman's rank corr.					Kendall's τ rank corr.				
Parameter	m	M	r_{alg}	d_{alg}	$\frac{r_{alg}}{d_{alg}}$	m	M	r_{alg}	d_{alg}	$\frac{r_{alg}}{d_{alg}}$	m	M	r_{alg}	d_{alg}	$\frac{r_{alg}}{d_{alg}}$
x1	-0.03	**-0.26**	**-0.21**	0.13	**-0.29**	0.02	**-0.26**	**-0.25**	**0.31**	**-0.34**	0.01	**-0.17**	**-0.18**	**0.22**	**-0.24**
x2	-0.07	**-0.31**	- **0.23**	0.13	**-0.21**	0.03	**-0.26**	**-0.24**	0.14	**-0.20**	0.02	**-0.18**	**-0.16**	0.10	**-0.14**
x3	0.14	0.08	-0.01	-0.12	0.12	-0.05	0.03	0.02	**-0.21**	***0.18***	-0.05	0.01	0.01	**-0.14**	***0.12***
x4	0.04	***-0.16***	***-0.16***	0.13	-0.09	-0.03	**-0.20**	***-0.18***	0.06	-0.11	-0.03	- ***0.12***	***-0.11***	0.04	-0.07

performs badly. Thus, the number of classes significantly influences the behavior of an algorithm, regardless of the family of the algorithm. Second, the number of features (x_2) provides a minor reduction of sensitivity to noise variation (resulting from low correlation to d_{alg}). This conclusion is also supported by the negative influence on $\frac{r_{alg}}{d_{alg}}$, r_{alg}. We also note that the number of features affects the maximal performance M, which shows (rather contrary to intuition) that more features may negatively affect performance in a noise-free scenario. This is most probably related to features that are not essentially related to the labeling process, thus inducing feature noise. Third, there is a correlation between the number of instances (x_3) and $\frac{r_{alg}}{d_{alg}}$. This shows that larger datasets (providing more instances) reduce sensitivity to noise variation. Last, fractal dimensionality (x_4) of a dataset has low, but statistically significant negative influence on M and on r_{alg}. Fractal dimensionality is indicative of the "complexity" of the dataset. Thus, if the dataset is complex (high x_4) machine learning is difficult even at low noise levels. We note that low r_{alg} may be preferable in cases where the algorithm should be stable even for low signal-to-noise ratios.

The correlation analysis demonstrates the connection between dataset characteristics and SRF dimensions. Consequently, the SRF can be used to reveal a-priori the properties of an algorithm with respect to a dataset of certain characteristics. This allows an expert to select a good algorithm for a given setting, based on the requirements of that settings. Such requirements may, e.g., relate to the stability of an algorithm in varying levels of noise and the expected maximum performance in non-noisy datasets.

5 Conclusions

Machine learning algorithms are often used in noisy environments. Therefore, it is important to know a-priori the properties of an algorithm with respect to a dataset of certain characteristics. In this work, we investigate whether some simple dataset properties (namely, number of classes, number of features, number of instances and fractal dimensionality) can help in the above direction.

We propose the "Sigmoid Rule" Framework, which describes a set of dimensions that may be used by a decision maker to choose a good classifier, or to estimate SRF dimensions, based on a range of dataset characteristics. Our approach is applicable to user modeling tasks, when the user changes behavior over time, and to any concept drift problems for data series mining. We showed that the parameters related to the behavior of learners correlate with dataset characteristics, and the range of their variation may be predicted using regression models. Therefore, SRF is a useful meta-learning framework, applicable to a wide range of settings that include noise. However, using these SRF models for parameter prediction does not provide enough precision to be used for performance estimation.

As part of our ongoing work, we examine whether the "Sigmoid Rule" also stands in the case of sequential classification. Preliminary experimental results on the "Climate" UCI dataset (taking into account its temporal aspect) indicate that, indeed, the "Sigmoid Rule" and therefore SRF are directly applicable, and can be used as a means to represent the behavior of an HMM-based classifier in the presence of noise. This finding may open the way to a broader use of the SRF, including sequential learners.

Acknowledgements. This research was partially supported by FP7 EU IP project KAP (grant agreement no. 260111).

References

1. Ali, S., Smith, K.A.: On learning algorithm selection for classification. Applied Soft Computing 6(2), 119–138 (2006)
2. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30(7), 1145–1159 (1997)
3. Camastra, F., Vinciarelli, A.: Estimating the intrinsic dimension of data with a fractal-based method. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(10), 1404–1407 (2002)
4. Chevaleyre, Y., Zucker, J.-D.: Noise-tolerant rule induction from multi-instance data. In: De Raedt, L. (ed.) Proceedings of the ICML 2000 Workshop on Attribute-Value and Relational Learning: Crossing the Boundaries (2000)
5. Cohen, W.W.: Fast effective rule induction. In: ICML (1995)
6. de Sousa, E., Traina, A., Traina Jr., C., Faloutsos, C.: Evaluating the intrinsic dimension of evolving data streams. In: Proceedings of the 2006 ACM Symposium on Applied Computing, pp. 643–648. ACM (2006)
7. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
8. Giannakopoulos, G., Palpanas, T.: Adaptivity in entity subscription services. In: ADAPTIVE (2009)
9. Giannakopoulos, G., Palpanas, T.: Content and type as orthogonal modeling features: a study on user interest awareness in entity subscription services. International Journal of Advances on Networks and Services 3(2) (2010)
10. Giannakopoulos, G., Palpanas, T.: The effect of history on modeling systems' performance: The problem of the demanding lord. In: ICDM (2010)
11. Giraud-Carrier, C., Vilalta, R., Brazdil, P.: Introduction to the special issue on meta-learning. Machine Learning 54(3), 187–193 (2004)
12. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD Explorations Newsletter 11(1), 10–18 (2009)

13. Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann (2006)
14. Kalapanidas, E., Avouris, N., Craciun, M., Neagu, D.: Machine learning algorithms: a study on noise sensitivity. In: Proc. 1st Balcan Conference in Informatics, pp. 356–365 (2003)
15. Keerthi, S.S., Shevade, S.K., Bhattacharyya, C., Murthy, K.R.K.: Improvements to platt's smo algorithm for svm classifier design. Neural Computation 13(3), 637–649 (2001)
16. Kuh, A., Petsche, T., Rivest, R.L.: Learning time-varying concepts. In: NIPS, pp. 183–189 (1990)
17. Li, Q., Li, T., Zhu, S., Kambhamettu, C.: Improving medical/biological data classification performance by wavelet preprocessing. In: Proceedings ICDM Conference (2002)
18. Pendrith, M., Sammut, C.: On reinforcement learning of control actions in noisy and non-markovian domains. Technical report, School of Computer Science and Engineering, The University of New South Wales, Sydney, Australia (1994)
19. Teytaud, O.: Learning with noise. Extension to regression. In: Proceedings of International Joint Conference on Neural Networks, IJCNN 2001, vol. 3, pp. 1787–1792. IEEE (2002)
20. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Academic Press (2003)
21. Wolpert, D.: The existence of a priori distinctions between learning algorithms. Neural Computation 8, 1391–1421 (1996)
22. Wolpert, D.: The supervised learning no-free-lunch theorems. In: Proc. 6th Online World Conference on Soft Computing in Industrial Applications. Citeseer (2001)
23. Wolpert, D.H.: The lack of a priori distinctions between learning algorithms. Neural Computation 8, 1341–1390 (1996)
24. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A.F.M., Liu, B., Yu, P.S., Zhou, Z.-H., Steinbach, M., Hand, D.J., Steinberg, D.: Top 10 algorithms in data mining. Knowl. Inf. Syst. 14(1), 1–37 (2008)

Building Decision Trees for the Multi-class Imbalance Problem

T. Ryan Hoens[1], Qi Qian[2], Nitesh V. Chawla[1], and Zhi-Hua Zhou[2]

[1] Department of Computer Science and Engineering
University of Notre Dame, Notre Dame IN 46556, USA
{thoens,nchawla}@cse.nd.edu
[2] National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210046, China
{qianq,zhouzh}@lamda.nju.edu.cn

Abstract. Learning in imbalanced datasets is a pervasive problem prevalent in a wide variety of real-world applications. In imbalanced datasets, the class of interest is generally a small fraction of the total instances, but misclassification of such instances is often expensive. While there is a significant body of research on the class imbalance problem for binary class datasets, multi-class datasets have received considerably less attention. This is partially due to the fact that the multi-class imbalance problem is often much harder than its related binary class problem, as the relative frequency and cost of each of the classes can vary widely from dataset to dataset. In this paper we study the multi-class imbalance problem as it relates to decision trees (specifically C4.4 and HDDT), and develop a new multi-class splitting criterion. From our experiments we show that multi-class Hellinger distance decision trees, when combined with decomposition techniques, outperform C4.4.

1 Introduction

One of the fundamental problems in data mining classification problems is that of class imbalance. In the typical binary class imbalance problem one class (negative class) vastly outnumbers the other (positive class). The difficulty of learning under such conditions lies in the induction bias of most learning algorithms. That is, most learning algorithms, when presented with a dataset in which there is a severely underrepresented class, ignore the minority class. This is due to the fact that one can achieve very high accuracy by always predicting the majority class, especially if the majority class represent 95+% of the dataset [3].

The multi-class classification problem is an extension of the traditional binary class problem where a dataset consists k classes instead of two. While imbalance is said to exist in the binary class imbalance problem when one class severely outnumbers the other class, extended to multiple classes the effects of imbalance are even more problematic. That is, given k classes, there are multiple ways for class imbalance to manifest itself in the dataset. One typical way is there is one "super majority" class which contains most of the instances in the dataset.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 122–134, 2012.

Another typical example of class imbalance in multi-class datasets is the result of a single minority class. In such instances $k-1$ instances each make up roughly $1/(k-1)$ of the dataset, and the "minority" class makes up the rest.

The multi-class imbalance problem is therefore interesting for two important reasons. First, as before, most learning algorithms do not deal with the wide variety of challenges multi-class imbalance presents. Secondly, a number of classifiers do not easily extend to the multi-class domain.

As a result, researchers have sought to exploit theoretical and empirical performance benefits of binary approaches for the multi-class problem. One common technique to do so is to decompose the multi-class problems into a set of binary class problems. This enables users to learn binary class classifiers on each of the subproblems which can then be combined into an ensemble in order to solve the multi-class problem. Such examples include "One-Versus-All" (OVA) and "Error Correcting Output Codes" (ECOC) [7].

One important distinction between an ensemble created using a decomposition technique, and a traditional ensemble in the binary class literature, is no single classifier in the decomposition ensemble can classify an instance in the multi-class domain. Thus while we use the word ensemble in this paper, we do not compare against traditional ensemble techniques (e.g., bagging [1], and AdaBoost [9]) as they are outside the scope of this paper. In order to avoid confusion, ensembles built using decomposition techniques will be known as "decomposition ensembles".

Contributions. While the multi-class imbalance problem is a serious problem in data mining, there has been little study on the effectiveness of decision trees on the multi-class imbalanced learning problem. Recently Hellinger distance decision trees (HDDTs) have been proposed as a way of solving the class imbalance problem for decision trees without sampling. Building upon this method, we propose a modified HDDT algorithm which improve its performance on multi-class datasets, along with an analytic result to explain the relative weaknesses of HDDT in the multi-class domain. We then demonstrate the effectiveness of various decomposition techniques on improving the performance of decision trees (both C4.4 and HDDT). We then specifically demonstrate how these techniques exploit the nature of HDDT on binary imbalanced datasets to build decomposition ensembles of HDDT classifiers which outperform other decision tree methods on two widely used metrics. Finally, we provide recommendations for building decision trees for the multi-class imbalance problem.

2 Methods

We apply a variety of methods to better understand the performance of decision trees in the class imbalance problem. Due to space restrictions, we limit the study to two popular decomposition techniques (OVA and ECOC), as well as building single trees.

2.1 Decomposition Techniques

As previously discussed, decomposition techniques have become a powerful tool in the data mining community to transfer (less studied) multi-class problems into (more studied) binary class problems. When considering decomposition techniques, an important factor is the size of the generated decomposition ensemble. Since one of the criteria when selecting decomposition methods to consider was the amount of computation time required, we selected two techniques which (generally) generate vastly different sized decomposition ensembles (and thus require vastly different computation time).

One-Versus-All Decomposition. The OVA technique is one of the simplest and most natural techniques for decomposing the multi-class problem into multiple binary class problems. In OVA, given c classes, c classifiers are built such that each one considers one of the classes to be the "positive" class while the remainder are combined into a "negative" class. When a new instance is seen, each classifier returns a probability estimate for the instance. An overall probability estimate is then obtained by combining each of the individual probability estimates into a vector of length c, and normalizing.

One of the main advantages of the OVA technique is that it is conceptually simple. Rifkin and Klautau [16] argue that this simplicity, combined with its superior performance, make OVA a very desirable technique which should be considered over its more complicated alternatives.

Error Correcting Output Codes Decomposition. ECOC is another popular method developed by Dietterich and Bakiri [7], which uses the concept of error correcting codes to learn a decomposition ensemble of classifiers. The choice of error correcting codes is a natural one as, assuming the codewords have hamming distance d, a maximum of $\lfloor \frac{d-1}{2} \rfloor$ errors can be made by the decomposition ensemble before misclassification occurs. This is a strong guarantee, and allows users to customize the size of the decomposition ensemble based on how many errors they expect versus the size of the codewords which they will allow.

More specifically, in ECOC each class is given an n-bit binary string called a "codeword". These codewords are generated such that the hamming distance between all codewords is maximized. Let c be an $m \times n$ matrix (where m is the number of classes), such that c_{ij} denote the jth bit for the codeword of class i. Given this, we can now learn a decomposition ensemble of n classifiers. For each classifier, the positive and negative classes are determined by the jth column of c. That is, if $c_{ij} = 1$, then class i is considered part of the positive class in classifier j. Otherwise, if $c_{ij} = 0$, class i is considered part of the negative class.

One of the most important considerations when building an ECOC decomposition ensemble is the length of the codewords. The maximum codeword length is $2^{m-1} - 1$. While building decomposition ensembles of this size results in the one most robust to errors, it also requires the most training time. Specifically, for 11 classes this method requires building a decomposition ensemble of size 1024.

While given the computing power available today this is of reasonable size, as the number of classes grows the problem quickly becomes intractable. Since having so many classes is rare in practice, and does not in fact occur for any datasets in this paper, we build codewords of maximum size for all datasets.

2.2 Decision Trees

Decision trees are one of the fundamental learning algorithms in the data mining community. The most popular of decision tree learning algorithm is C4.5 [14]. Recently Hellinger distance decision trees (HDDTs) [4] have been proposed as an alternative method for building decision trees for binary class datasets which exhibit class imbalance.

Provost and Domingos [13] recommend a modification to C4.5 known as C4.4. In C4.4 decision trees are constructed by building unpruned and uncollapsed C4.5 decision trees which use Laplace smoothing at the leaves. These choices are due to empirical results [13] demonstrating that a fully built unpruned, uncollapsed tree with Laplace smoothing outperforms all other configurations, and thus are used in all experiments in this paper.

The important function to consider when building a decision tree is known as the *splitting criterion*. This function defines how data should be split in order to maximize performance. In C4.4 this function is gain ratio, which is a measure of purity based on entropy [14], while in HDDT this function is Hellinger distance. In the next section we motivate Hellinger distance as a splitting criterion, and then subsequently devise a strategy for improving its performance on multi-class datasets.

Hellinger Distance Splitting Criterion. Hellinger distance is a distance metric between probability distributions used by Cieslak and Chawla [4] to create Hellinger distance decision trees (HDDTs). It was chosen as a splitting criterion for the binary class imbalance problem due to its property of skew insensitivity. Hellinger distance is defined as a splitting criterion as [4]:

$$d_H(X_+, X_-) = \sqrt{\sum_{j=1}^{p} \left(\sqrt{\frac{|X_{+j}|}{|X_+|}} - \sqrt{\frac{|X_{-j}|}{|X_-|}} \right)^2} \tag{1}$$

where X_+ is the set of all positive examples, X_- is the set of all negative examples and X_{+j} (X_{-j}) is the set of positive (negative) examples with the jth value (of p distinct values) of the relevant feature.

Since Hellinger distance defines the distance between probability distributions, it does not naturally extend to the multi-class problem. This is in contrast to gain ratio — which is based on entropy — which is easily extensible to any number of classes. Specifically, since Hellinger distance is a distance metric, any natural extension would be attempting to determine the distance between c probability distributions, where c is the number of classes. Since this is not a well defined problem, we propose an extension to the HDDT algorithm for the multi-class problem.

Algorithm 1. *Calc_Multi_Class_Hellinger*

```
Require: Training set $T$, Feature $f$, Set of classes $C$
Let Hellinger $\leftarrow -1$.
Let $V_f$ be the set of values of feature $f$.
for each pair of subsets of $C$: $C_1 \subset C$, $C_2 = C \setminus C_1$: do
    for each value $v \in V_f$ do
        Let $w \leftarrow V_f \setminus v$
        cur_value $\leftarrow (\sqrt{|T_{f,v,+}|/|T_+|} - \sqrt{|T_{f,v,-}|/|T_-|})^2 + (\sqrt{|T_{f,w,+}|/|T_+|} - \sqrt{|T_{f,w,-}|/|T_-|})^2$
        if $cur\_value >$ Hellinger then
            Hellinger $\leftarrow cur\_value$
        end if
    end for
end for
return $\sqrt{\text{Hellinger}}$
```

Multi-Class HDDT. In order to overcome the shortcomings of Hellinger distance as a splitting criterion for the multi-class problem, we employ techniques similar to the decomposition algorithms described in Section 2.1. That is, given the set of classes C, we consider each unique pair of subsets: $C_1 \subset C$, $C_2 = C \setminus C_1$ and consider all classes in C_1 as the positive class, and all classes in C_2 as the negative class[1]

Algorithm 1 outlines the approach to incorporating Hellinger distance in learning multi-class decision trees. Let T_C indicate the subset of training set T which has its class in set C, and $T_{k,j,C}$ identifies the subset which has its class in set C and has value j for feature k.

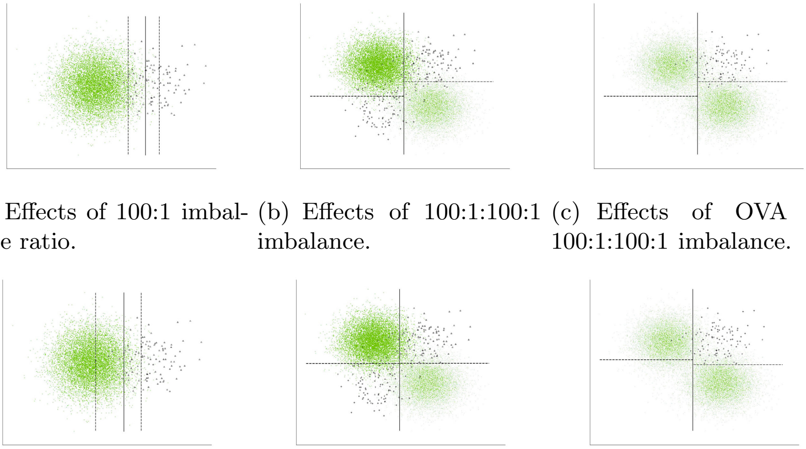
The important aspect of this version of the Hellinger distance splitting criterion is the reduction of the multiple classes into all relevant binary class possibilities. This choice enables Hellinger distance to try find the best split between all possible choices of positive and negative class, and thus any meaningful split available to it in the multi-class domain.

This distance calculator can then be used as the splitting criterion in a decision tree algorithm in order to build multi-class HDDTs (MC-HDDTs). Comparing MC-HDDTs further to HDDT shows us that for the binary class problem, exactly the same tree will be learned as the original version. *Our algorithm can therefore be recommended in lieu of HDDT, as it returns the same tree for the binary case while offering better performance on the multi-class problem.*

3 Analysis of the Splitting Criteria

One of the major research questions in this paper is why the performance of HDDT suffers in the multi-class case when compared to C4.4, especially in light

[1] Note that two pairs of subsets (e.g., (C_1, C_2) and (D_1, D_2)) are considered equal if $C_1 = D_1$ or $C_1 = D_2$.



(a) Effects of 100:1 imbalance ratio.

(b) Effects of 100:1:100:1 imbalance.

(c) Effects of OVA on 100:1:100:1 imbalance.

(d) Effects of 100:1 imbalance.

(e) Effects of 100:1:100:1 imbalance.

(f) Effects of OVA on 100:1:100:1 imbalance.

Fig. 1. Comparison of the effects of various class distributions on the ability of information gain (top) and Hellinger distance (bottom) to correctly determine the class boundary which optimizes AUC

of their performances on binary class imbalanced datasets. In this section we present an analytic example which demonstrates how HDDT and C4.4 behave when a binary class problem is transformed into a multi-class problem and then back again. Due to space limitations we limit ourselves to a single example which demonstrates an example of Hellinger distance performing poorly in the multi-class case.

For our analytic example we created a simulated dataset with 4 classes, with centers on the corners of a square, such that their means were separated by 2σ. In the upper left and lower right corners, we simulated 10,000 examples, while in the lower left and upper right corners we simulated only 100 examples. This gives us a class imbalance ratio of 10,000:100:10,000:100 (C.V.: 0.98). We then decomposed the 4 class problem into a binary class problem by removing the lower half of the square (as depicted in Figure 3). In order to determine their performance, each of the experiments was run 100 times, and the (W)AUROC (defined in Section 4) computed.

Figures 1(a) and 1(d) are representative examples of the effects of gain ratio and Hellinger distance (respectively) on the binary class problem. From the splits, we see that Hellinger distance is much more aggressive when splitting into the majority class. When considering their performance, we see that, based on AUROC, HDDT wins 85 out of the 100 runs. This increase in performance is therefore an effect of Hellinger distance aggressively attempting to capture as much of the minority class as possible, while gain ratio remains very conservative.

In the multi-class case (Figures 1(b) and 1(e)) Hellinger distance once again is very aggressive in attempting to capture as much of the minority class as possible, while C4.4 is much more conservative. Due to the nature of this problem, however, the more conservative approach is better able to capture the multi-distributional aspect of the problem. This is demonstrated by the fact that, based on WAUROCs, C4.4 wins 82 of the 100 runs. Thus, for multi-class, Hellinger distance is not able to adequately separate the two classes, instead being overwhelmed by the spurious information from the extra classes.

In order to better understand this phenomena, consider the right-most horizontal split Hellinger makes in the multi-class case. For this split, Hellinger distance considers the "top" points to be the positive class and the "bottom" points to be the negative class. As evidenced by the inaccuracy of the top left points, Hellinger distance is not able to accurately partition the space. Gain ratio, on the other hand is able to arrive at a better split point which more accurately represents the boundary for this problem.

Finally we consider the case of OVA decomposition on the dataset. Figure 1(f) shows Hellinger distance is very good at capturing the minority class. This favorable splitting is exactly what would be expected from such a binary class imbalanced dataset, and thus explains the performance increase HDDT sees over C4.4 when used in conjunction with OVA. This hypothesis is further confirmed when we note that HDDT obtains a higher AUROC in 80 of the 100 runs, thus confirming that it is the preferred classifier to use.

Given these results, we now better understand the dynamics of Hellinger distance in the binary class problem which result in inferior performance in the multi-class domain. Further research into overcoming these challenges might prove useful in developing a single decision tree approach which, without sampling, is able to outperform the others in the case of multi-class imbalance.

4 Experiments

We implemented MC-HDDT in WEKA [10], and used WEKA's built-in OVA and ECOC to train each of the classifiers. In order to make fair comparisons, we split the experiments into three separate categories, namely: single trees, OVA decomposition, and ECOC decomposition. This separation is done to highlight the difference in performance of HDDT and C4.4 under different decomposition techniques. That is, by comparing each method within a category, we are providing a fair comparison of the different decision tree techniques.

Table 1 gives the relevant simple statistics about the datasets used in this paper. One of the main goals when choosing the datasets to consider was ensuring that they were imbalanced. To measure imbalance in multi-class datasets, we use the "coefficient of variation" (C.V.) as recommended by Cieslak and Chawla [5]. Specifically, C.V. is the proportion of the deviation in the observed number of examples for each class versus the expected number of examples in each class. In this paper we consider datasets with a C.V. above 0.35 – a class ratio of 2:1 on a binary dataset – imbalanced. This leaves us with the 17 datasets listed.

4.1 Configuration

In order to ensure a fair comparison of the methods, we ran 50 iterations [15] of 2-fold cross-validation. We chose 2-fold cross-validation due to the small number of instances of some classes in the datasets. Due to space restrictions, we only consider weighted area under the receiver operating characteristic (WAUROC) [19]. We chjse this metrics as it is a commonly used criterion when comparing classifiers in the multi-class imbalance case.

Table 1. Statistics for the datasets used in this paper. C.V. is the coefficient of variation, # Ftrs is the number of features, and # Insts is the number of instances.

Dataset	C.V.	# Ftrs	# Insts	# Classes
abalone	0.711	9	4177	4
artificial	0.594	8	5109	5
auto-mpg	0.621	8	398	3
balance-scale	0.541	5	625	3
bgp	1.260	9	24984	4
car	1.082	7	1728	4
connect-4	0.714	43	67557	3
dermatology	0.455	35	366	6
dna	0.394	180	3186	3
glass	0.761	9	214	6
page-blocks-5	1.747	10	5473	5
sat	0.372	36	6435	6
segment	0.535	20	2310	3
solar-flare-2	0.535	12	1066	6
splice	0.393	61	3190	3
vehicle	0.370	19	846	3
yeast	1.005	9	1484	9

4.2 Statistical Tests

While many different techniques have been applied to attempt to compare classifier performance across multiple datasets, Demšar suggests comparisons based on ranks. We follow this recommendation and rank the performance of each classifier by its average performance, with 1 being the best. Since we seek to determine whether or the HDDT methods are statistically significantly better than the existing methods, we use the Friedman and Bonferroni-Dunn tests as was recommended by Demšar [6].

The Friedman test is first applied to determine if there is a statistically significant difference between the rankings of the classifiers. That is, it tests to see if the rankings are not merely randomly distributed. Next, as recommended by Demšar, we preform the Bonferroni-Dunn test to compare each classifier against the control classifier.

4.3 Results

As stated previously we break the experiment into three different categories. Each of the categories corresponds to a different level of computational effort required to construct the classifier, with single trees requiring the least amount

Table 2. WAUROC values for the various methods over each of the datasets. Bold numbers indicate overall best performance. The number in parenthesis indicates the rank in the category. A ✓ indicates that the method performs statistically significantly worse than the other method in its category at the relevant confidence level.

	Single Tree		OVA		ECOC	
Dataset	C4.4	MC-HDDT	C4.4	HDDT	C4.4	HDDT
abalone	0.71484 (1)	0.71007 (2)	0.73751 (2)	0.74073 (1)	**0.74869 (1)**	0.74803 (2)
artificial	0.87548 (2)	0.87932 (1)	0.87913 (2)	0.88178 (1)	0.90344 (2)	**0.90474 (1)**
auto-mpg	0.90093 (2)	0.90806 (1)	0.91153 (2)	0.91476 (1)	0.91153 (2)	0.91476 (1)
balance-scale	0.90607 (1)	0.90495 (2)	0.90486 (2)	0.90651 (1)	0.90486 (2)	**0.90651 (1)**
bgp	0.80268 (1)	0.79698 (2)	0.81378 (2)	0.81414 (1)	0.82418 (2)	**0.82446 (1)**
car	0.97662 (2)	0.98652 (1)	0.98209 (2)	0.99244 (1)	0.98262 (2)	**0.99413 (1)**
connect-4	0.87879 (1)	0.85156 (2)	0.88971 (1)	0.87900 (2)	**0.88971 (1)**	0.87900 (2)
dermatology	0.97794 (2)	0.98283 (1)	0.98357 (2)	0.99079 (1)	0.99594 (2)	**0.99682 (1)**
dna	0.97508 (1)	0.96680 (2)	**0.98436 (1)**	0.98232 (1)	0.98436 (1)	0.98232 (2)
glass	0.80585 (1)	0.79370 (2)	0.83867 (2)	0.84393 (1)	0.88161 (2)	**0.88348 (1)**
page-blocks-5	0.98104 (2)	0.98111 (1)	0.98446 (2)	0.98480 (1)	0.98731 (2)	**0.98940 (1)**
sat	0.96262 (1)	0.96124 (2)	0.97273 (2)	0.97471 (1)	0.98679 (2)	**0.98715 (1)**
segment	0.98656 (2)	0.98848 (1)	0.99437 (2)	0.99650 (1)	0.99437 (2)	0.99650 (1)
solar-flare-2	0.91886 (2)	0.92032 (1)	0.91683 (2)	0.91856 (1)	**0.92098 (1)**	0.92035 (2)
splice	0.97459 (2)	0.97476 (1)	0.98457 (1)	0.98262 (2)	0.98457 (1)	0.98262 (2)
vehicle	0.95696 (2)	0.96139 (1)	0.97164 (2)	0.97717 (1)	0.97164 (2)	0.97717 (1)
yeast	0.73156 (1)	0.71541 (2)	0.76973 (2)	0.77137 (1)	0.80843 (2)	**0.80871 (1)**
Avg. Rank	1.52941	1.47059	1.82353	1.17647	1.70588	1.29412
$\alpha = 0.05$			✓			
$\alpha = 0.10$			✓		✓	

of work, and ECOC requiring the most. For the sake of space, however, the WAUROC values for each of the methods is presented in Table 2.

Table 2 also contains the results of the statistical test described in Section 4.2. A classifier receives a check mark if it is considered statistically significantly worse than the best classifier (i.e., the classifier with the lowest average rank) in its category (e.g., single tree, OVA, ECOC) at the noted confidence level.

Single Tree Performance. When considering the single tree performances, C4.4 and MC-HDDT perform equivalently. This is an interesting result for multi-class imbalanced data sets, and further corroborates the intuition established with the illustrations in Section 3. As discussed, this is mainly due to the aggressive nature of the splits which Hellinger distance tries to create. The consequence of this analysis is further evidenced in the OVA performance.

Hellinger distance, as a criterion, is limited in capturing the multi-class divergences. Nevertheless, we recommend MC-HDDT as a decision tree classifier, as it reduces to HDDT for binary class datasets (achieving statistically significantly superior performance over C4.4 [4]), and is a competitive alternative to C4.4 for multi-class datasets (no statistically significant variation in performance).

OVA Performance. When considering OVA performance, HDDT significantly outperforms C4.4. This result confirms our understanding of the binary class performances of each of the classifiers. That is, when decomposing the multi-class problem into multiple binary problems, the binary class problems obtained are (often) extremely imbalanced. This fact is further exacerbated by the fact that the multi-class dataset itself is highly imbalanced.

Thus in the OVA approach, each binary classifier in the decomposition ensemble must deal with the class imbalance problem. Since HDDT has been shown to perform statistically significantly better than C4.4 in this scenario, we expect to see HDDT outperforming C4.4 when using the OVA approach. Based on the observations obtained, we can conclude that our intuition is correct and, furthermore, that when using OVA decomposition for multi-class imbalance, HDDT are the appropriate decision tree learning to choose.

ECOC Performance. When comparing the relative performance of the classifiers, we see that HDDT outperforms C4.4 almost as well as in the OVA approach. While the statistical significance is only $\alpha = 0.10$, we see that it is not statistically significant at the $\alpha = 0.05$ threshold by one dataset. As Table 2 shows, some of the performance differences were quite small. Thus it seems reasonable to believe that with more datasets we might see the same statistical significance with this method as was shown in OVA, as we would expect the same performance gains of using HDDT over C4.4 in this case as well.

This expectations of better performance of HDDT over C4.4 is due to similar reasoning as the OVA case. That is, by decomposing the problems into multiple binary problems, the class imbalance will still be a major concern. However, the ECOC approach will result in $2^{m-1} - 1$ binary datasets. Some of these will be highly imbalanced, while others may be balanced depending on the respective class distributions. Nevertheless, HDDT is able to capitalize with ECOC. It is able to achieve stronger separability on highly imbalanced combinations, and achieves comparable performance to C4.4 on the relatively balanced class combinations, and thus, as a collective, it is able to outperform C4.4.

Overall Performance. When considering the overall performance of each method as given in Table 2, we see that, in general, the more computational power used, the better the performance. That is, the ECOC methods outperform the OVA methods which outperform the single tree methods.

This is an unsurprising result, as a wealth of data mining literature demonstrates that combining a large number of classifiers into an ensemble is a powerful technique for increasing performance. The decomposition ensemble techniques employed in this paper are also of particular interest, as the diversity of the classifiers created in the decomposition ensembles is quite high. That is, since the class values under consideration are changing between datasets, the classifiers are not merely learning on different permutations of the underlying instances, instead having the decision boundaries themselves change. It is well known that diversity is important to creating good ensembles [11].

5 Related Work

A number of methods have been proposed to counter the class imbalance issue, however a large portion has focused on the binary class problem. Sampling methods have emerged as a de facto standard, but present numerous challenges when being extended to multiple classes. This is due to the complexity arising from the combination of multiple class imbalance types, different amounts of sampling, different sampling methods, and different cost matrices. Thus to apply any reasonable optimization criteria to discovering the optimal sampling amount is computationally prohibitive.

Rescaling [8] is a general method for cost-sensitive and class-imbalance problems which changes the distribution of the original data [20]. As there are many methods that can change the distribution, rescaling can be realized in numerous ways (e.g., by sampling, instance-weighting, threshold moving, etc.). Sampling is a widely used rescaling method to deal with the class-imbalance problem. The method balances the distribution modifying the training set to either increase the presence of the minority class (e.g., random oversampling, SMOTE [2]), or reduce the majority class (e.g., undersampling). Another popular rescaling method is instance-weighting. In this method, instead of removing or adding instances, a weight is generated for each instance according to its misclassification cost, which is passed to a cost-blind classifier which uses instance weights [17]. A final common approach is threshold moving, wherein the decision threshold is modified in order to achieve the minimal cost in cost-sensitive learning [8,21].

Cost sensitive learning methods have been developed to deal with the different costs of misclassification [18]. For example, the cost of misclassifying a cancer patient as healthy is much higher than the cost of misclassifying a healthy patient as having cancer. Given this, cost sensitive problems require the minimization of the misclassification cost rather than misclassification errors. The class-imbalance problem can thus be considered a cost-sensitive problem where the costs are unequal and unknown [12]. Most cost-sensitive learning methods are actually based on rescaling [20], and therefore it is natural that by assigning the appropriate misclassification cost for each class, cost-sensitive approaches can be used to deal with the class-imbalance problems.

6 Conclusion and Discussion

In this paper we compared different methods of building C4.4 and Hellinger distance decision trees for multi-class imbalanced datasets. Given the different amounts of computation time required by each method, we investigated the problem in three separate categories: single tree, OVA, and ECOC.

In the single tree case we found that MC-HDDT performs comparably to C4.4. While MC-HDDT does not statistically significantly outperform C4.4, it is a reasonable alternative to C4.4 for all classification problems. This is an important result as it gives practitioners another viable tool to use when confronted with a new dataset.

Alternatively, when the analysis was extended to build decomposition ensembles of binary classifiers HDDT became the clear choice. Given the skew insensitivity of Hellinger distance as a splitting criterion, coupled with the nature of skew in the resulting problems, the gains in performance become significant. With this in mind, we recommend HDDT for all multi-class imbalanced learning when used in a decomposition method.

Another important observation stems from the overall performance. Specifically we see that the more complex (and thus more computationally intensive) algorithms give real gains in performance. We can therefore revise our recommendation, this time recommending the use of HDDT in an ECOC decomposition ensemble if the user has enough computational power. Otherwise, the user should consider an OVA decomposition ensemble with HDDT, and, finally, if not enough computational power exists for such a decomposition, building MC-HDDTs. We recommend MC-HDDTs over C4.4, as even though the difference between them is not statistically significant for multi-class datasets, MC-HDDT reduces to HDDT for binary class datasets, where it has been demonstrated to be strongly skew insensitive and statistically significantly over C4.4. As a result, MC-HDDT may be considered the recommended decision tree algorithm.

Finally, Section 3 illustrated the challenges Hellinger distance faces in the multi-class domain. With this understanding further research can now explore the problem of multi-class Hellinger distance and attempt to overcome the demonstrated difficulties to provide a robust classifiers for multi-class problems.

Acknowledgements. Work is supported in part by the NSF Grant ECCS-0926170, NSFC (60903103, 61021062) and the Notebaert Premier Fellowship.

References

1. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
2. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. JAIR 16, 321–357 (2002)
3. Chawla, N.V.: Data mining for imbalanced datasets: An overview. In: Data Mining and Knowledge Discovery Handbook, pp. 875–886 (2010)
4. Cieslak, D.A., Chawla, N.V.: Learning Decision Trees for Unbalanced Data. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 241–256. Springer, Heidelberg (2008)
5. Cieslak, D.A., Chawla, N.V.: Start globally, optimize locally, predict globally: Improving performance on imbalanced data. In: ICDM, pp. 143–152 (2008)
6. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. JMLR 7, 30 (2006)
7. Dietterich, T., Bakiri, G.: Error-correcting output codes: A general method for improving multiclass inductive learning programs. In: AAAI, pp. 395–395 (1994)
8. Domingos, P.: Metacost: A general method for making classifiers cost-sensitive. In: KDD, pp. 155–164 (1999)
9. Freund, Y., Schapire, R.: A Desicion-Theoretic Generalization of On-Line Learning and an Application to Boosting. In: Vitányi, P.M.B. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995)
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Exp. News. 11(1), 10–18 (2009)
11. Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation, and active learning. In: NIPS, pp. 231–238 (1995)
12. Maloof, M.A.: Learning when data sets are imbalanced and when costs are unequal and unknown. In: ICML WLIDS (2003)
13. Provost, F., Domingos, P.: Tree induction for probability-based ranking. Machine Learning 52(3), 199–215 (2003)

14. Quinlan, J.R.: Induction of decision trees. Machine Learning 1(1), 81–106 (1986)
15. Raeder, T., Hoens, T., Chawla, N.: Consequences of Variability in Classifier Performance Estimates. In: ICDM, pp. 421–430 (2010)
16. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. JMLR 5, 101–141 (2004)
17. Ting, K.M.: An instance-weighting method to induce cost-sensitive trees. TKDE 14(3), 659–665 (2002)
18. Turney, P.D.: Types of cost in inductive concept learning. In: ICML, pp. 15–21 (2000)
19. Van Calster, B., Van Belle, V., Condous, G., Bourne, T., Timmerman, D., Van Huffel, S.: Multi-class auc metrics and weighted alternatives. In: IJCNN, pp. 1390–1396 (2008)
20. Zhou, Z.-H., Liu, X.-Y.: On multi-class cost-sensitive learning. In: AAAI, pp. 567–572 (2006)
21. Zhou, Z.-H., Liu, X.-Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. TKDE 18(1), 63–77 (2006)

Scalable Random Forests for Massive Data

Bingguo Li, Xiaojun Chen, Mark Junjie Li, Joshua Zhexue Huang,
and Shengzhong Feng

Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
Shenzhen 518055, China
{bg.li,xj.chen,jj.li,zx.huang,sz.feng}@siat.ac.cn

Abstract. This paper proposes a scalable random forest algorithm SRF with MapReduce implementation. A breadth-first approach is used to grow decision trees for a random forest model. At each level of the trees, a pair of map and reduce functions split the nodes. A mapper is dispatched to a local machine to compute the local histograms of subspace features of the nodes from a data block. The local histograms are submitted to reducers to compute the global histograms from which the best split conditions of the nodes are calculated and sent to the controller on the master machine to update the random forest model. A random forest model is built with a sequence of map and reduce functions. Experiments on large synthetic data have shown that SRF is scalable to the number of trees and the number of examples. The SRF algorithm is able to build a random forest of 100 trees in a little more than 1 hour from 110 Gigabyte data with 1000 features and 10 million records.

Keywords: MapReduce, Random forests, Histogram.

1 Introduction

Data with millions of records and thousands of features present a big challenge to current data mining algorithms. On one hand, it is difficult to build classification models from such massive data with serial algorithms running on single machines. On the other hand, most classification algorithms are not capable of building accurate models from extremely high dimensional data with thousands of features. However, such high dimensional massive data exist in many application domains, such as text mining, bio-informatics and e-commerce.

Random forests [1] is an effective ensemble model for classifying high dimensional data [2]. A random forest consists of K decision trees, each grown from a data set randomly sampled from the training data with replacement. At each node of a decision tree, a subset of m features is randomly selected and the node is split according to the m features. Breiman [1] suggested $m = Log_2(M) + 1$ where M is the total number of features in data. For very high dimensional data, M is very big and m is much smaller than M. Therefore, decision trees in a random forest are grown from subspaces of features [3][4][5]. The random forest classifies data according to the majority votes of individual decision trees. Due to the computation

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 135–146, 2012.

of a large number of decision trees, it is extremely difficult to build random forest models from large data sets with millions of records on single servers.

MapReduce [6][7][8] is a simple programming model for distributed computing. It abstracts away many low-level details such as task scheduling and data management, and conceptualizes a computational process as a sequence of map and reduce functions. In a map phase, the same map function is dispatched to all computing nodes and executes the same set of operations in parallel. In the reduce phase, the reduce function merges results from different mapper tasks. If the data file is large, more data blocks are created and distributed on more computing nodes. The map and reduce functions are automatically dispatched to more computing nodes. Therefore, MapReduce model is scalable to large data.

To implement the random forest algorithm in MapReduce, a straightforward method is to build one decision tree from each data block. This assumes that each data block is one training data sampled from the large training data set. The decision tree algorithm is implemented in one map function so it is dispatched to all computing nodes to build all decision trees from the local data blocks. The majority voting is performed in the reduce function. Such implementation is adopted in Apache Mahout[1]. Two main drawbacks in this simple implementation are: 1) The data blocks are hard partitions of the large training data and can have biased distributions different from the training data. This problem results in weak and biased trees; 2) As the map function builds a decision tree recursively, it can cause memory leakage if the tree is large and complex.

In this paper, we propose a new scalable random forest algorithm (SRF) for constructing random forest models from massive data. The SRF algorithm takes advantages of MapReduce programming model to gain high scalability on large data. Instead of using the recursive process to grow trees, a breadth-first tree growing method is adopted. Starting from the root nodes, all trees grow on level basis. The nodes of all trees on each level split up into children nodes in one pair of map and reduce functions. A random forest model is built with a sequence of D pairs of map and reduce functions where D is the depth of the highest tree in the forest. To split nodes on each level, one pair of map and reduce functions are used. The histograms of features for each node are calculated first in the map function and all sets of histograms for the same node in a tree are merged into one set of global histograms in the reducer job. We employ the method in SPDT [9] to calculate the histograms from each local data block and merge them into the global histograms to split the node. The reducer also calculates the node split function such as information gain and determines the split conditions of data to generate the children nodes. At start, a reference table is generated to record the data sets for different trees that are randomly sampled from the training data with replacement (bagging).

We have conducted a series of experiments on both synthetic and real-life data sets and compared SRF with SPDT and random forests in Mahout. The result showed that SRF obtained higher accuracy than SPDT and accuracy of SRF was similar to the random forests in Mahout. To further compare SRF with Mahout, we

[1] http://mahout.apache.org

added noise to OCR data set. On this noise data, the accuracy of random forests in Mahout reduced to 51.9% while the accuracy of SRF was 78.6%. On a separate sparse data Real-sim[2], Mahout random forests was not able to produce a model and generated stack overflow error message due to memory leakage but SRF worked fine. We used a large synthetic data with more than one thousand attributes and millions of records, to test the scalability of SRF. With 30 computing nodes, SRF was able to build a random forest with 100 trees in a little more than 1 hour from a massive data set of 110 Gigabytes with 1000 features and 10 million records. This was indeed a significant result. SRF also demonstrated a linear property with respect to number of trees and number of examples.

The rest of the paper is organized as follows. Section 2 gives a brief review of the SPDT algorithm. In Section 3, we present the SRF algorithm in details. We present experiment results on real-life data and scalability tests in Section 4. The paper is concluded in Section 5.

2 Related Work

Building decision trees is the major function of building a random forest. Traditional decision algorithms [10][11] use a recursive process to create a decision tree from a training data set. These algorithms will have problems if the training data or the tree is too big to fit in the main memory. Scalable decision tree algorithms have been proposed to handle large data. Some take an approach to pre-sort the training data before building the decision tree, such as SLIQ [12], SPRINT [13] and ScalParC [14]. Others compute the histograms of attributes and split the training data according to the histograms, such as BOAT [15], CLOUDS [16], SPIES [17] and SPDT [9]. The later are more scalable as the tree growing process is no longer relevant to the size of training data after all histograms are created. The creation of histograms can be easily parallelized. Google also proposed PLANET [18] for regression trees based on MapReduce programming model. PLANET only supports sampling without replacement.

In this work, we use a breadth-first method to construct decision trees for a random forest. We select the streaming parallel decision tree algorithm SPDT recently developed at IBM as a framework to develop the breadth-first tree growing process. Figure 1 sketches the process of the SPDT algorithm. It runs in a distributed environment with one master node and several workers. Each worker stores $1/W$ percentage of the data where W is the number of workers. To grow a decision tree, the master node instructs workers to compute the local histograms of features from their local data blocks. After local histograms are complete, the workers send them to the master which merges them into the global histograms. The global histograms are then used to compute the conditions to split the nodes and grow the decision tree. After the nodes in the same level are split, the master node instructs the workers again to compute histograms for the newly generated children nodes. This process continues until no node needs further split.

[2] http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/binary.html

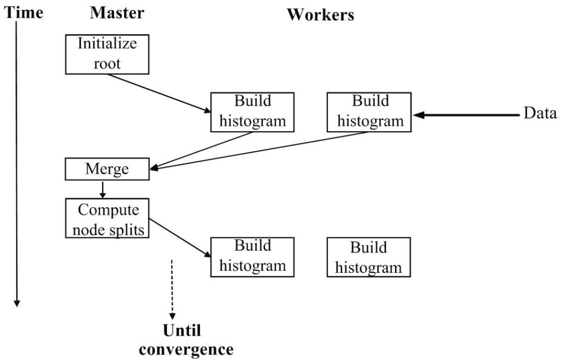


Fig. 1. The parallel framework of the SPDT algorithm

3 Scalable Random Forest Algorithm

In this section, we present a scalable random forest algorithm that uses SPDT node split method to grow decision trees in building a random forest model. The process of random forest building is implemented in the MapReduce programming model and runs on a distributed cloud computing platform.

3.1 Breadth-First Random Forest Construction

In a distributed environment, we create K decision trees for a random forest in parallel. Instead of the recursive process, we use the breadth-first method to create multiple decision trees. For example, to create three decision trees A, B, C in Figure 2, we first generate three root nodes. Then, we compute the best splits of the three root nodes in parallel and generate 6 children nodes, two from each root. We continue this process to generate grand children nodes, great grand children nodes, and etc. At each node, if the pre-defined stop conditions meet, the node is treated as a leaf node and no further split is necessary. After all leaf nodes are found, the tree growing process terminates and a random forest is obtained.

3.2 Scalable Random Forest Algorithm

With MapReduce programming model, we not only distribute map and reduce functions to create decision trees, but also partition the training data into data blocks and distribute them on different computing nodes. To create sample data sets for decision trees, we create an index table, called bagging table with K columns, each recording the examples that were selected in the sample data for a decision tree with the method of sampling with replacement. This bagging table is also distributed together with data blocks.

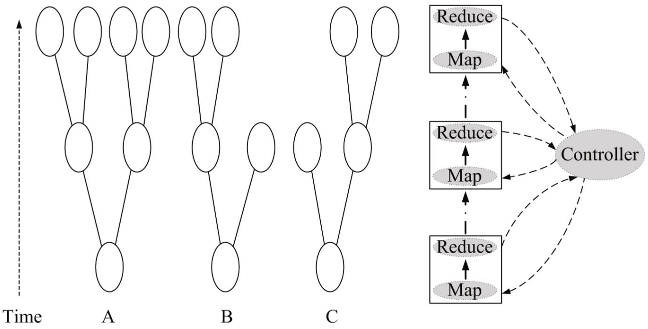


Fig. 2. Breadth-first method to grow decision trees for a random forest

The scalable random forest algorithm is composed of a sequence of map and reduce functions. Each mapper and reducer iteration creates one set of nodes in the same level of K trees, as shown in Figure 2. In the distributed cloud environment, each computing node stores a fixed number of data blocks. Each data block is accessed by one mapper dispatched by the controller that runs on the master node. The mapper is used to create the local histograms of subspace features for all nodes in the current level for all K trees. The split conditions of a node in a tree and the bagging table are used to select the objects that belong to the node. At a root node, no split condition exists and only the bagging table tells which objects in a data block are selected for the tree to be created.

After all local histograms are computed by all mapper jobs, they are sorted on the tree ids. The local histograms for the same tree are sent to the same reducer to compute the global histograms at each node. The reducer also calculates the best split from the global histograms and send the best split conditions back to the controller to update the random forest model. After a pair of map and reduce functions completed, each decision tree grows a new level of nodes, which are taken as the current level of nodes for the next pair of map and reduce functions. If a node satisfies the stop condition, it is marked as the leaf node.

3.3 Mapper, Reducer and Controller

The pseudocode of the mapper procedure is described in Algorithm 1. Each record of the data block is checked against the bagging table $\mathscr{T}$. If it belongs to the sample data for the tree, it is used to compute the local histograms of subspace features. After scanning all the training set,the local histograms for all decision trees are obtained.

At the end of the map phase, the mapper sends all local histograms in a set of (key, value set) pairs to reducers. The key is the id of a decision tree and the value is local histograms of nodes in that tree. In this way, we ensure all local histograms for the same decision tree are sent to one reducer.

Algorithm 1. Mapper Procedure

```
Input:
- $D^*$ : the training dataset;
- $\mathscr{M}$ : the random forests model, it contains the random forests constructed
  so far, nodes in the current level of all trees contain selection conditions of
  objects;
- $\mathscr{T}$ : the bagging table, it records the bagging indexes of each tree.

Output: Generate histogram tuples $hist\text{-}tup$ for nodes of all trees in the
  current level;
Method:
for each record $(x, y) \in D^*$ do
    for all decision tree $k \in \mathscr{M}$ do
        if tree $k$ has nodes to build AND the record is specified for tree $k$ in
          bagging table $\mathscr{T}$ then
            $hist\text{-}tup_k \leftarrow update(record, num)$ //update histograms for tree $k$
        end if
    end for
end for
for all decision tree $k \in \mathscr{M}$ do
    Output(treeId of $k$, $hist\text{-}tup_k$) // output pairs of tree id and histogram
      tuples
end for
```

The pseudocode of the reducer procedure is described in Algorithm 2. The reducer receives pairs of (key, value set) from different mappers. The key values are sorted ids of decision trees and the value sets are the local histograms of the trees from all mappers. The reducer merges the value sets of local histograms into global histograms with respect to the tree id and the node. The reducer computes the best split conditions from the global histograms for the node. If the split does not justify a split, the node is marked as a leaf node and no further split will occur. Otherwise, new children nodes are created under the node and the split conditions are recorded in node tuples for the children nodes. After all histograms are processed, the reducer sends the pairs of (id, *node-tuple*) to the controller. The controller adds the new nodes to the random forest model with the split conditions.

The pseudocode of the controller procedure is described in Algorithm 3. The controller starts with initialization of an empty model $\mathscr{M}$. Vector $\mathbb{N}$ is used to record the number of nodes to split in a level of each tree. $\mathbb{N}$ is initialized with each tree having one root node. $\mathbb{N} = [1, 1, 1]$ in the root level of Figure 2. Each iteration of a map and reduce pair generates a new $\mathbb{N}$ recording the number of nodes for each tree in the newly split level. For example, $\mathbb{N} = [2, 2, 2]$ in the level above the root and $\mathbb{N}$ becomes $[4, 2, 2]$ afterwards. $\mathbb{N} = [0, 0, 0]$ after the last level.

Algorithm 2. Reducer Procedure

```
Input:
- $k$ : the id of tree;
- $\mathbb{V}$ : the value set, receiving from different mappers.

Output: Calculate the best split conditions at nodes for decision trees;
Method:
$hist\text{-}tup_k \leftarrow merge(\mathbb{V})$ // merge local histograms into global ones
for all built node $i$ in tree $k$ do
    The histogram of node $i$ is $hist\text{-}tup_k(i)$
    if $hist\text{-}tup_k(i)$ satisfies stop condition then
        $node\text{-}tup_i \leftarrow leaf$
    else
        $candidSplits \leftarrow Uniform(hist\text{-}tup_k(i))$ // compute the best split conditions
        $node\text{-}tup_i \leftarrow split$
    end if
end for
Output($k$, $node\text{-}tup$)
```

The loop of the controller checks whether $\mathbb{N}$ contains non-zero elements and terminates if all elements in $\mathbb{N}$ are zeros. Inside the loop, the controller configures a MapReduce job first and then dispatches it to the computing nodes to perform a pair of map and reduce functions on one level of nodes of all decision trees. The controller also passes the information of the nodes in the current level to each mapper for computing local histograms. The controller calculates the number of reducers for the MapReduce job according to the number of nodes in $\mathbb{N}$ to balance the load to reducers. After all reducers complete, parseOutput function processes the results from reducers and generates the new node information in *tree-tup*. The *tree-tup* data is used to update the random forest model $\mathscr{M}$ and computes a new $\mathbb{N}$ to record the number of new nodes generated in each tree.

4 Experiments

In this section we show the classification results of SRF on large complex data sets and comparisons of them with those by SPDT and Mahout. We also demonstrate the scalability of SRF to very large data sets. The results show the capability of SRF in building random forest models from extremely large data with 10 million records and 1000 features in less than 2 hours. Such models would be very difficult, if not impossible, to build via traditional random forest algorithms.

4.1 Data Sets

Two data sets were used in experiments. The first set was used to test the classification performance of SRF in accuracy and compare classification results

Algorithm 3. Controller Procedure

```
Input:
- $\mathscr{M}$ : the random forests model, $\mathscr{M} = \phi$;
- $\mathbb{N}$ : the vector, $\mathbb{N} = [1, 1, ...]$.

Output: Construct a random forests model with MapReduce;
Method:
while $\mathbb{N}$ has non-zero elements do
    ConfigureMapReduce(job)
    Dispatch(job)
    $tree\text{-}tup = parseOutput(job)$
    $\mathbb{N} = updateForests(\mathscr{M}, tree\text{-}tup)$
end while
```

with those by SPDT and Mahout. Four real-life data sets were selected and these data sets were used in evaluating SPDT in [9]. The characteristics of these data sets are summarized in Table 1. They can be downloaded from UCI repository and Pascal Large Scale Learning Challenge[3].

Table 1. Real-life data sets used in accuracy experiments

Dataset	#Features	#Train Set	#Test Set	%Classes
Isolet	617	6,238	1,559	26
Multiple Features	649	2000	2000	10
Face Detection	900	1,000,000	100,000	2
OCR	1,156	1,000,000	100,000	2

The second set contained four synthetic data sets that were generated for scalability test of SRF. The four synthetic sets are listed in Table 2. The points in the same class in these data sets have Gaussian distributions. To generate separable classes in the synthetic data, we specified several central points with labels first and calculated the distance between a generated record and the central points. We set the class label of the record as the label of the central point that had the minimum distance to the record.

4.2 Experiment Settings

The experiment environment was composed of 30 machines, each having 8 cpus and 15 GB memory running CentOS Linux operating system. Hadoop was installed on these machines to form a MapReduce runtime environment. Each machine was configured with 8 mappers and 8 reducers. The size of data block was 64 MB by default.

[3] ftp://largescale.ml.tu-berlin.de/largescale

Table 2. Characteristics of synthetic sets for scalability evaluation

Dataset	#Features	#Train Set	#Classes	%Size of data(GB)
D1	1,000	10,000,000	5	110
D2	1,200	7,000,000	10	91.6
D3	1,400	5,000,000	15	76
D4	1,600	2,000,000	20	32.4

50 bins were used to build histograms. In performance experiment, a random forest had 100 decision trees and the size of subspace at each node was $log_2(M)+1$, where M was the number of features in training data.

In scalability experiment, we investigated the scalability of SRF with respect to four factors, i.e., the number of decision trees, the size of data, the size of data block and the number of machines. For the number of decision trees, we started with 1 single tree and added to 20 trees, then increased 20 more trees other times. For size of data, we started with 200,000 examples and increased the size of data with 200,000 more examples each time. For size of data block, we started with 16 MB of data block and double the size of data block each time. For the number of machines, we started with 15 machines, and increased 5 more machines each time.

4.3 Performance Results

Table 3 lists the classification results in term of accuracy of SPDT, Mahout and SRF on the four real-life data sets described in Table 1. We can see that the accuracies of SRF were higher than that of SPDT on all the four data sets. The most prominent result was from the OCR data set that had 1156 features and one million records. The increase of accuracy over SPDT was 19%. This result demonstrated that SRF could get better performance than SPDT in handling massive and high dimensional data, as SRF is a random forests algorithm, it can get better performance than decision tree algorithm (SPDT) in handling massive data.

Table 3. Accuracies of SPDT, Mahout and SRF on four real-life data sets

Dataset	#Acc.(%) of SPDT	#Acc.(%) of Mahout	#Acc.(%) of SRF
Isolet	77.42	92.6	92.8
Multiple Features	91.5	98.5	97
Face Detection	96.69	91.5	97.47
OCR	60.65	78.9	79.5

For Mahout and SRF, the accuracy of SRF was closely similar with Mahout random forests on Isolet and Multiple Features training data sets. The reason is that the size of these two training data sets was no more than 64 MB, Mahout and SRF constructed all decision trees based on one data block, thus these two algorithms degenerated into traditional random forests algorithms. On the other hand, SRF could obtained higher accuracy than Mahout random forests on massive data, such as Face Detection training set.

To illustrate the fact that random forests in Mahout builds all decision trees on the first data block when the number of blocks is larger than the number of decision trees, we added a block size of noise records, which was generated randomly for two labels, in front of the OCR training set. The accuracy of random forests in Mahout decreased rapidly from 78.9% to 51.9% while the accuracy of SRF decreased from 79.5% to 78.6% on the noised OCR training set. The reason is that, for Mahout, all decision trees were built on the first data block, which contained the added noise data. As a result, SRF outperforms Mahout in handling massive data.

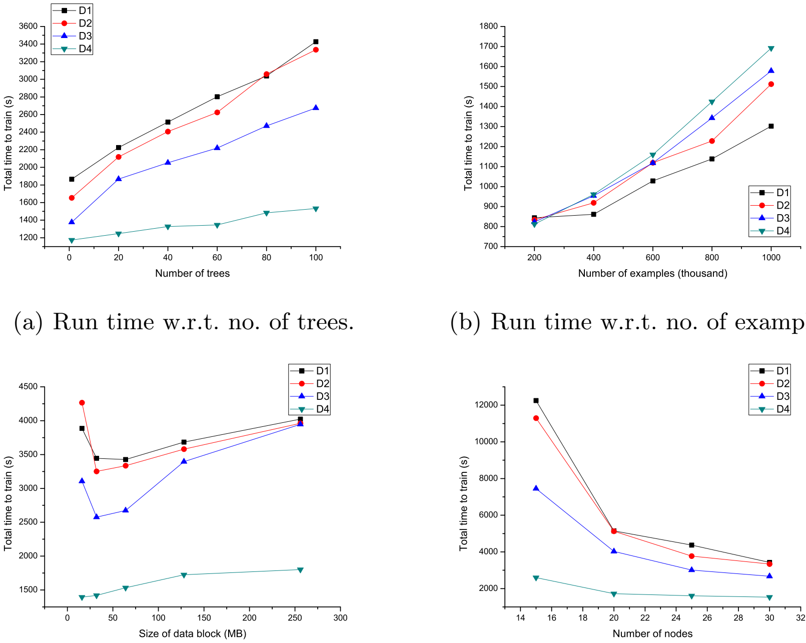
In addition, Mahout may lead to memory leakage problem while handling massive data. For example, Mahout random forests generated stack overflow error message when dealing with Real-sim training data set, which has 35,000 examples and 20,958 features. As random forests in Mahout built decision trees with depth-first mode, which may cause memory leakage problem.

4.4 Scalability

Figure 3 shows the scalability on four synthetic data sets with respect to the number of trees in random forests, the number of examples in data, the size of data block and the number of machines used. Figure 3(a) shows that the time used to build a random forest model increased linearly as the number of trees increased in the model. The run times for building one tree model for data sets D1-D4 are 1174s, 1377s, 1654s and 1866s respectively. The run time increases slowly as more trees are added to the model. For instance in data set D4, only less than 4 extra seconds were added to the total run time when each additional tree was added to the model. The larger the data set, more time it takes when more trees are added. However, the speed of time increase is very slow. This result demonstrates that SRF is scalable to the number of trees in the model.

Figure 3(b) shows the scalability of run time on four synthetic sets with respect to the number of examples in data. We can see a linear increase in time as more examples were added in building random forest models. When the number of examples was small, the differences of run times for the four data sets were very small. As more examples were involved, the run time increased but very slow. The large the data size, the larger the increase of run time. However, the speed of increase was not fast. This demonstrates that SRF is also scalable to the data size.

Figure 3(c) shows the change of run time over the change of the size of data block. The size of data block had impact on the run time of SRF. On the one hand, the smaller data block generates more mappers. However, if the number of mappers exceeds the mapper capacity of the system, the run time will increase rapidly. On the other hand, the larger data block generates heavy load mapper. From the chart, we can see that the proper size of data block is 32MB or 64MB for the data sets.



(a) Run time w.r.t. no. of trees.

(b) Run time w.r.t. no. of examples.

(c) Run time w.r.t. data block size.

(d) Run time w.r.t. no. of nodes.

Fig. 3. Scalability results on large synthetic data sets

Figure 3(d) shows the scalability of SRF with respect to the number of machines involved. We can see that the run time dropped rapidly as more machines added. This demonstrates that SRF is able to handle very large data by adding more machines.

5 Conclusions

We have presented the scalable random forest algorithm SRF and its implementation in MapReduce. In the algorithm, we adopted the breadth-first approach to build decision trees in a sequence of pairs of map and reduce functions to avoid the memory leakage in the depth-first recursive approach and make the algorithm more scalable. We have demonstrated the scalability of SRF with very large synthetic data sets and the results have shown SRF's ability in building random forest models from data with millions of records. Our future work is to further optimize SRF in the area of load balance to make it more efficient and scalable.

Acknowledgements. This research is supported in part by Shenzhen New Industry Development Fund under grant No.CXB201005250021A.

References

1. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
2. Banfield, R., Hall, L., Bowyer, K., Kegelmeyer, W.: A comparison of decision tree ensemble creation techniques. IEEE Transactions on Pattern Analysis and Machine Intelligence, 173–180 (2007)
3. Ho, T.: Random decision forests. In: Proceedings of the Third International Conference on Document Analysis and Recognition, vol. 1, pp. 278–282. IEEE (1995)
4. Ho, T.: C4.5 decision forests. In: Proceedings of Fourteenth International Conference on Pattern Recognition, vol. 1, pp. 545–549. IEEE (1998)
5. Ho, T.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)
6. White, T.: Hadoop: The definitive guide. Yahoo Press (2010)
7. Venner, J.: Pro Hadoop. Springer (2009)
8. Lam, C., Warren, J.: Hadoop in action (2010)
9. Ben-Haim, Y., Tom-Tov, E.: A streaming parallel decision tree algorithm. The Journal of Machine Learning Research 11, 849–872 (2010)
10. Breiman, L.: Classification and regression trees. Chapman & Hall/CRC (1984)
11. Quinlan, J.: C4.5: Programs for machine learning. Morgan Kaufmann (1993)
12. Mehta, M., Agrawal, R., Rissanen, J.: Sliq: A Fast Scalable Classifier for Data Mining. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) EDBT 1996. LNCS, vol. 1057, pp. 18–32. Springer, Heidelberg (1996)
13. Shafer, J., Agrawal, R., Mehta, M.: Sprint: A scalable parallel classifier for data mining. In: Proceedings of the International Conference on Very Large Data Bases, pp. 544–555. Citeseer (1996)
14. Joshi, M., Karypis, G., Kumar, V.: Scalparc: A new scalable and efficient parallel classification algorithm for mining large datasets. In: Proceedings of the First Merged International and Symposium on Parallel and Distributed Processing, Parallel Processing Symposium, IPPS/SPDP 1998, pp. 573–579. IEEE (1998)
15. Gehrke, J., Ganti, V., Ramakrishnan, R., Loh, W.: Boatoptimistic decision tree construction. In: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, pp. 169–180. ACM (1999)
16. AlSabti, K., Ranka, S., Singh, V.: Clouds: Classification for large or out-of-core datasets. In: Conference on Knowledge Discovery and Data Mining (1998)
17. Jin, R., Agrawal, G.: Communication and memory efficient parallel decision tree construction. In: 3rd SIAM International Conference on Data Mining, San Francisco, CA (2003)
18. Panda, B., Herbach, J., Basu, S., Bayardo, R.: Planet: massively parallel learning of tree ensembles with mapreduce. Proceedings of the VLDB Endowment 2(2), 1426–1437 (2009)

Hybrid Random Forests: Advantages of Mixed Trees in Classifying Text Data

Baoxun Xu[1], Joshua Zhexue Huang[2], Graham Williams[2], Mark Junjie Li[2], and Yunming Ye[1]

[1] Department of Computer Science, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China
[2] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
amusing002@gmail.com, zx.huang@siat.ac.cn, Graham.Williams@togaware.com, jj.li@siat.ac.cn, yeyunming@hit.edu.cn

Abstract. Random forests are a popular classification method based on an ensemble of a single type of decision tree. In the literature, there are many different types of decision tree algorithms, including C4.5, CART and CHAID. Each type of decision tree algorithms may capture different information and structures. In this paper, we propose a novel random forest algorithm, called a hybrid random forest. We ensemble multiple types of decision trees into a random forest, and exploit diversity of the trees to enhance the resulting model. We conducted a series of experiments on six text classification datasets to compare our method with traditional random forest methods and some other text categorization methods. The results show that our method consistently outperforms these compared methods.

Keywords: Random Forests, Hybrid Random Forest, Classification, Decision Tree.

1 Introduction

Random forests [1,2] are a popular classification method which builds an ensemble of a single type of decision tree. The decision trees are often either built using C4.5 [3] or CART [4], but only one type was exploited within a single random forest. In recent years, random forests have attracted increasing attention due to (1) its competitive performance compared with other classification methods, especially for high-dimensional data, (2) algorithmic intuitiveness and simplicity, and (3) its most important capability - "ensemble" using bagging [5] and stochastic discrimination [2].

The most popular forest construction procedure was proposed by Breiman [1], novelly using bagging to generate training data subsets for building individual trees. A subspace of features is then randomly selected at each node to grow branches of a decision tree. The trees are then combined as an ensemble into a random forest [1].

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 147–158, 2012.

In the literature, different types of decision trees algorithms have been proposed, including C4.5, CART and CHAID [6]. Each type of decision tree algorithms employs a different tree building process and captures different discriminative information.

Random forests gain some of their performance advantage through the diversity of the trees in the resulting ensemble. We can add another kind of diversity to the random forest framework by removing any potential bias from using a single type of decision tree. We propose to use several different types of decision trees for each training data subset, and select the best tree as the individual tree classifier in the random forest model.

Our method is motivated by the experiences of foresters in dealing with the development and care of hybrid forests. An important concept in Forestry is that of a "hybrid forest." Such a forest uses multiple tree species as a mixed planting in accordance with soil structure (moisture, nutrients, acidity). This method has demonstrated highly economic, ecological and practical value in forestry research. Mimicing this idea, we have developed a hybrid random forest method to explore whether we can further enhance the classification performance of a random forest ensemble classifier. Specifically, we build three different types of tree classifiers (C4.5, CART and CHAID) for each training data subset. We then evaluate the performance of the three classifiers and select the best tree. In this way, we build a hybrid random forest which may include different types of decision trees in the ensemble. The added diversity of the decision trees can effectively improve the accuracy of each tree in the forest, and hence the accuracy of the ensemble.

To demonstrate the effectiveness of our proposed method, we apply it to the popular application of text classification. With the ever-increasing volume of text data from the Internet, databases, and archives, text categorization has become a key technique for handling and organizing text data. It has received growing attention in recent years. A set of popular and mature machine learning approaches have been deployed for categorizing text documents, including random forests [8], support vector machines (SVM) [9], naive Bayes (NB) [10], k-nearest neighbors (KNN) [11], and decision trees. Due to algorithmic simplicity and prominent classification performance for high dimensional data, random forests have become a preferred method.

In this paper, we compare the performance of our random forest with that of other three random forest methods, i.e., C4.5 random forest, CART random forest and CHAID random forest, and other three mainstream text categorization methods, i.e., support vector machines, naive Bayes and k-nearest neighbors, on six datasets. The experimental results show that our hybrid random forest achieves improved classification performance over these six compared methods.

The rest of this paper is organized as follows. Section 2 introduces the framework for building a hybrid Random Forest, and gives a brief analysis of the method. The evaluation methods are presented in Section 3, we present experimental results in Section 4. Our conclusions and future work are presented in Section 5.

2 Hybrid Random Forests

In this section, we introduce a general framework for building hybrid random forests. We then briefly review three types of decision tree algorithms, i.e., C4.5, CART and CHAID. We also present our hybrid random forest algorithm that integrates the best trees from the different types of decision tree algorithms.

2.1 Framework for Building Hybrid Random Forest

As an ensemble learner, the performance of a random forest is highly dependent on two factors: the diversity among the trees and the accuracy of each tree [12]. Diversity is commonly obtained by using bagging and random subspace sampling. We introduce a further element of diversity by using different types of trees.

Continuing our analogy with forestry, the different data subsets from bagging represents the "soil structures." Different decision tree algorithms represent "different tree species". Our approach has two key aspects: one is to use three types of decision tree algorithms to generate three different tree classifiers for each training data subset; the other is to evaluate the accuracy of each tree as the measure of tree importance. In this paper, we use the out-of-bag accuracy to assess the importance of a tree.

Following Breiman, we use bagging to generate a series of training data subsets from which we build trees. For each tree, the data subset used to grow the tree is called the "in-of-bag" (IOB) data and the remaining data subset is called the "out-of-bag" (OOB) data. Since OOB data is not used for building trees we can use this data to objectively evaluate each tree's accuracy and importance. The OOB accuracy gives an unbiased estimate of the true accuracy of a model.

Given n instances in a training dataset D and a tree classifier $h_k(IOB_k)$ built from the k'th training data subset IOB_k, we define the OOB accuracy of the tree $h_k(IOB_k)$ for each $di \in D$ as:

$$OOBAcc_k = \frac{\sum_{i=1}^{n} I(h_k(d_i) = y_i; d_i \notin IOB_k)}{\sum_{i=1}^{n} I(d_i \notin IOB_k)} \tag{1}$$

where I(.) is an indicator function. The larger the $OOBAcc_k$, the better classification quality a tree has.

We use the out-of-bag data subset OOB_i to calculate the out-of-bag accuracies of the three types of trees (C4.5, CART and CHAID) with evaluation values A_1, A_2 and A_3 respectively.

Fig. 1 illustrates the procedure for building a hybrid random forest model. Firstly, a series of IOB/OOB datasets are generated from the entire training dataset by bagging. Then, three types of tree classifiers (C4.5, CART and CHAID) are built using each IOB dataset. The corresponding OOB dataset is used to calculate the OOB accuracies of the three tree classifiers. Finally, we select the tree with the highest OOB accuracy as the final tree classifier, which is included in the hybrid random forest.

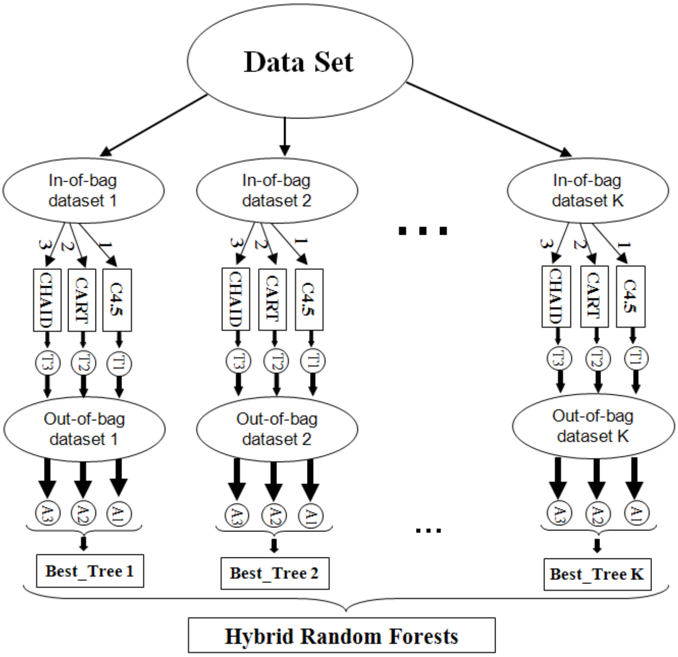


Fig. 1. The Hybrid Random Forests framework

Building a hybrid random forest model in this way will increase the diversity among the trees. The classification performance of each individual tree classifier is also maximized.

2.2 Decision Tree Algorithms

The core of our approach is the diversity of decision tree algorithms in our random forest. Different decision tree algorithms grow structurally different trees from the same training data. Selecting a good decision tree algorithm to grow trees for a random forest is critical for the performance of the random forest. Few studies have considered how different decision tree algorithms affect a random forest. We do so in this paper.

The common decision tree algorithms are as follows:

Classification Trees 4.5. (C4.5) is a supervised learning classification algorithm used to construct decision trees. Given a set of pre-classified objects, each described by a vector of attribute values, we construct a mapping from attribute values to classes. C4.5 uses a divide-and-conquer approach, which is similar to recursive partitioning, to grow decision trees. C4.5 selects the test that maximizes the information gain ratio (IGR) [3].

Classification and Regression Tree. (CART) is a recursive partitioning method that can be used for both regression and classification. Beginning with the entire dataset, a tree is constructed by splitting subsets of the dataset by considering all predictor variables for splitting. The best predictor is chosen at each node using a variety of impurity or diversity measures. The goal is to produce subsets of the data which are homogeneous with respect to the target variable [4]. The main difference between C4.5 and CART is the test selection and evaluation process.

Chi-squared Automatic Interaction Detector. (CHAID) method is based on the chi-square test of association. A CHAID decision tree is constructed by repeatedly splitting subsets of the space into two or more nodes. To determine the best split at any node, any allowable pair of categories of the predictor variables is merged until there is no statistically significant difference within the pair with respect to the target variable [6,7].

From these decision tree algorithms, we can see that the difference lies in the way to split a node, such as the split functions and binary branches or multi-branches. In this work we use these different decision tree algorithms to build a hybrid random forest.

2.3 Algorithm

In this subsection, we present our hybrid random forest algorithm which integrates the three types of tree classifiers. The detailed steps are introduced in

Algorithm 1.

In Algorithm 1, lines 10-17 loop to build K decision trees. In the loop, Line 11 samples the training data D by sampling with replacement to generate an in-of-bag data subset IOB_i for building a decision tree. Lines 12-15 build three types of tree classifiers (C4.5, CART and CHAID). In this procedure, Line 13 calls the function $createTree_j()$ to build a tree classifier. Line 14 calculates the out-of-bag accuracy of the tree classifier. After this procedure, Line 16 selects the tree classifier with the maximum out-of-bag accuracy. K decision trees are thus generated to form a hybrid random forest model $\mathscr{M}$.

Generically, function $createTree_j()$ first creates a new node. Then, it tests the stopping criteria to decide whether to return to the upper node or to split this node. If we chose to split this node, then we randomly select m features as a subspace for node splitting. These features are used as candidates to generate the best split to partition the node. For each subset of the partition, $createTree_j()$ is called again to create a new node under the current node. If a leaf node is created, it returns to the parent node. This recursive process continues until a full tree is generated.

3 Evaluation Methods

We use two measures to evaluate the classification performance of the hybrid random forest, the test accuracy and the F1 metric. The test accuracy measures the performance of a random forest on a separate test dataset. The F1 metric is a commonly used measure of classification performance.

Algorithm 1. Hybrid Random Forest Algorithm

```
Input:
- $D$ : the training dataset,
- $A$ : the features space $\{A_1, A_2, ..., A_M\}$,
- $Y$ : the class features space $\{y_1, y_2, ..., y_q\}$,
- $K$ : the number of trees,
- $m$ : the size of subspaces.

Output: A random forest $\mathscr{M}$;
Method:
for $i = 1$ to $K$ do
    draw a bootstrap sample in-of-bag data subset $IOB_i$ and out-of-bag data subset $OOB_i$ from training dataset D;
    for $j = 1$ to 3 do
        $h_{i,j}(IOB_i) = createTree_j()$;
        use out-of-bag data subset $OOB_i$ to calculate the out-of-bag accuracy $OOBAcc_{i,j}$ of the tree classifier $h_{i,j}(IOB_i)$ by Equation (1);
    end for
    select $h_i(IOB_i)$ with the highest out-of-bag accuracy $OOBAcc_i$ as best tree $i$;
end for
combine the $K$ optimal tree classifiers $h_1(IOB_1), h_2(IOB_2), ..., h_K(IOB_K)$ into a random forest $\mathscr{M}$

Function createTree()
create a new node $\mathscr{N}$;
if stopping criteria is met then
    return $\mathscr{N}$ as a leaf node;
else
    randomly select $m$ features as a subspace;
    use these $m$ features as candidates to generate the best split for the node to be partitioned;
    call createTree() for each split;
end if
return $\mathscr{N}$;
```

Test Accuracy. Let D_t be a test dataset and Y_t be the class labels. Given $d_i \in D_t$, the number of votes for d_i on class j is

$$N(d_i, j) = \sum_{k=1}^{K} I(h_k(d_i) = j) \tag{2}$$

The test accuracy is calculated as

$$Acc = \frac{1}{n} \sum_{i=1}^{n} I(N(d_i, y_i) - \max_{j \neq y_i} N(d_i, j) > 0) \tag{3}$$

where n is the number of objects in D_t and y_i indicates the true class of d_i.

F1 Metric. To evaluate the performance of classification methods in dealing with an unbalanced class distribution, we use the F1 metric introduced by Yang and Liu [13]. This measure is equal to the harmonic mean of recall (α) and precision (β). The overall F1 score of the entire classification problem can be computed by a micro-average and a macro-average.

Micro-averaged F1. This is computed globally over all classes, and emphasizes the performance of a classifier on common classes. Define α and β as follows:

$$\alpha = \frac{\sum_{i=1}^{q} TP_i}{\sum_{i=1}^{q}(TP_i + FP_i)}, \quad \beta = \frac{\sum_{i=1}^{q} TP_i}{\sum_{i=1}^{q}(TP_i + FN_i)} \tag{4}$$

where q is the number of classes. TP_i (True Positives) is the number of objects correctly predicted as class i, FP_i (False Positives) is the number of objects that are predicted to belong to class i but do not. The micro-averaged F1 is computed as:

$$MicroF1 = \frac{2\alpha\beta}{\alpha + \beta} \tag{5}$$

Macro-averaged F1. This is first computed locally over each class, and then the average over all classes is taken. It emphasizes the performance of a classifier on rare categories. Define α and β as follows:

$$\alpha_i = \frac{TP_i}{(TP_i + FP_i)}, \quad \beta_i = \frac{TP_i}{(TP_i + FN_i)} \tag{6}$$

$F1$ for each category i and the macro-averaged F1 are computed as:

$$F1_i = \frac{2\alpha_i\beta_i}{\alpha_i + \beta_i}, \quad MacroF1 = \frac{\sum_{i=1}^{q} F1_i}{q} \tag{7}$$

The larger the MicroF1 and MacroF1 values are, the better the classification performance of the classifier.

4 Experiments

In this section, we conduct experiments to demonstrate the effectiveness of the hybrid random forest algorithm for classifying text data. Text datasets with various sizes and characteristics are used in the experiments. The experimental results show that the hybrid random forest algorithm not only outperforms single-tree type random forest algorithms, i.e., C4.5_RF, CART_RF and CHAID_RF, in classification accuracy, but also outperforms other three mainstream text categorization methods, i.e., SVM, NB and KNN.

4.1 Datasets

In the experiments, we used six real-world text datasets. These text datasets are selected due to their diversities in the number of terms or features, the number of documents, and the number of classes. Their dimensionalities vary from 2000 to 11,465, numbers of instances vary from 918 to 11,162 and the minority class rate varies from 0.32% to 6.43%. In each text dataset, we randomly select 70% of documents as the training dataset, and the remaining data as the test dataset. Detailed information of the six text datasets is listed in Table 1.

Table 1. Summary statistic of 6 text datasets

Dataset	#Terms	#Documents	#Classes	%Minority class
Fbis	2000	2463	17	1.54
Re0	2886	1504	13	0.73
Oh5	3012	918	10	6.43
Re1	3758	1657	25	0.6
Wap	8460	1560	20	0.32
Ohscal	11465	11162	10	6.35

These datasets are frequently used as text document classification benchmark data [14]. Dataset **Fbis** was compiled from Foreign Broadcast Information Service TREC-5 [15]. The datasets **Re0** and **Re1** were selected from Reuters-21578 text categorization test collection Distribution 1.0 [16]. Datasets **Oh5** and **Ohscal** are from the OHSUMED subset of the MEDLINE database [17]. **Wap** is from the WebACE project (WAP) [18].

4.2 Test Accuracy Improvement

The purpose of this experiment is to evaluate the effect of the hybrid random forest method on accuracy. The six text datasets were analyzed and results were compared with other three random forest methods (C4.5_RF, CART_RF and CHAID_RF). For each text dataset, we ran each random forest algorithm against different sizes of feature subspaces. Since the number of features in these datasets was very large, we started with a subspace of 15 features and increased the subspace with 5 more features each time. For a given subspace size, we built 100 trees for each random forest model. In order to obtain a stable result, we built 80 random forest models for each subspace size, each dataset and each algorithm, and computed the averages of the test accuracy as the final result for comparison.

Fig. 2 shows the plots of the average test accuracy of the four random forest models in different sizes generated with the four methods from the six text datasets. For the same number of features, the higher the accuracy, the better the result. From these figures, we can observe that the hybrid random forest

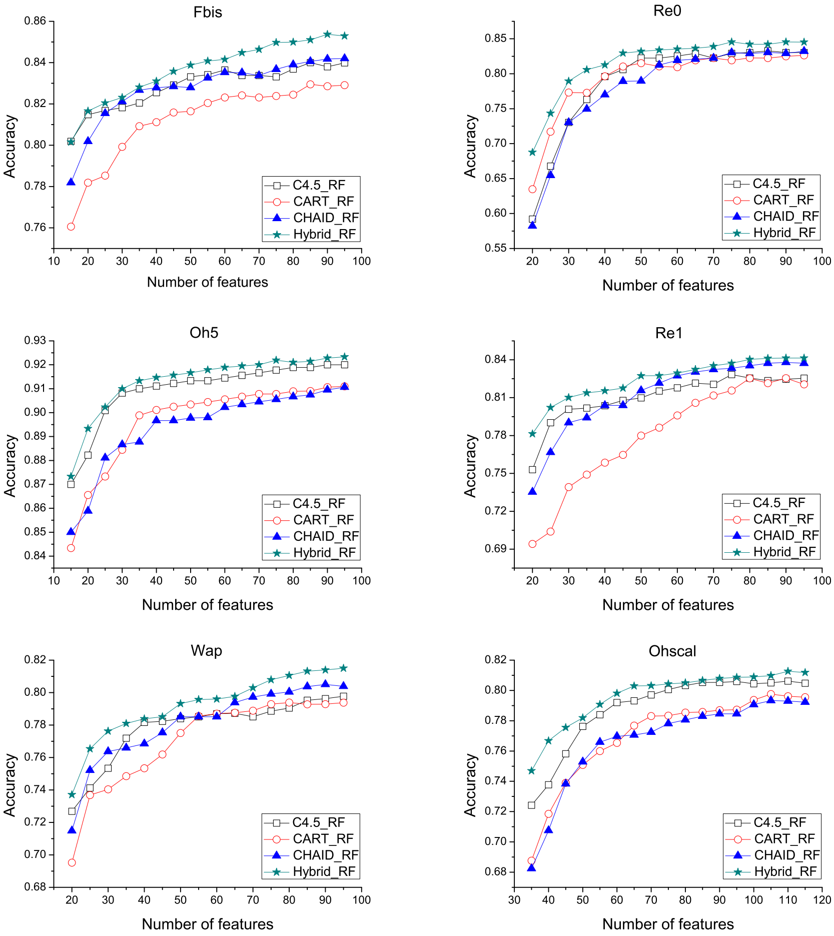


Fig. 2. Test accuracy changes against the number of features in the subspace on the 6 text datasets

algorithm consistently performs better than the other three random forest algorithms. The advantages are more obvious in the smaller subspaces. The hybrid random forest algorithm quickly achieves high accuracy as the subspace size increases. The other three random forest algorithms require larger subspaces to achieve a similar accuracy. These results illustrate that the hybrid random forest algorithm outperforms the other three random forest algorithms in the classification accuracy results on all the six text datasets.

To further investigate the performance of the hybrid random forest, we computed the average accuracy of the trees in each single-type random forest. This is compared to the average accuracy of the trees of the same type within the one hybrid random forest. In all comparisons, the subspace size of $\sqrt{M}$ features

Table 2. A comparison of the average accuracy of trees within a single-type random forest and the average accuracy of trees of that same type within the hybrid random forest

Name	C4.5		CART		CHAID	
	C4.5_RF	Hybrid_RF	CART_RF	Hybrid_RF	CHAID_RF	Hybrid_RF
Fbis	0.6379	**0.6489**	0.6102	**0.6414**	0.6382	**0.6536**
Re0	0.6132	**0.6324**	0.6271	**0.6478**	0.6171	**0.6304**
Oh5	0.6516	**0.6664**	0.6134	**0.6515**	0.6245	**0.6607**
Re1	0.6611	**0.6793**	0.6058	**0.6608**	0.6516	**0.6718**
Wap	0.5267	**0.5343**	0.5086	**0.5396**	0.5195	**0.5297**
Ohscal	0.5391	**0.5448**	0.4732	**0.5004**	0.4665	**0.5204**

was used, where M is the total number of features in the dataset. The results are shown in Table 2. For example, for tree type C4.5 and dataset Fbis, the average accuracy of all trees from the random forest built using C4.5 (named as C4.5_RF) is 0.6379. The average accuracy of all C4.5 trees from the hybrid random forest (named as Hybrid_RF) is 0.6489. It is clearly seen in Table 2 that tree classifiers of any given type in our hybrid random forest always have higher average classification accuracy than those using only trees of the same one type.

4.3 Performance Comparisons of other Text Classification Method

We conduct a further experimental comparison against other three widely used text categorization methods, i.e., support vector machines (SVM), Naive Bayes (NB), and k-nearest neighbor (KNN). The SVM uses a linear Kernel with a regularization parameter of 0.03125, which is often used in text categorization. For Naive Bayes, we adopted the multi-variate Bernoulli event model that is frequently used in text classification [19]. For k-nearest neighbor (KNN), we set the number of neighbors as 13. In the experiments, we use WEKA's implementation for these three text classification methods [20]. We use a single subspace size of 90 features in all the six datasets to run the random forest algorithms, which provides a consistent result as shown in Fig. 2. In order to obtain stable results, we built 20 random forest models for each random forest algorithm and each dataset, and present the average results, we can see that the range of values are less than ±0.005 and the hybrid trees are always more accurate.

The comparison results are listed in Fig. 3, 4 and 5. While the improvement is often quite small, there is always an improvement demonstrated. We observe that our proposed method always outperforms the compared mainstream text categorization methods.

5 Conclusion and Future Work

We have presented a new hybrid random forest algorithm which increases diversity amongst the ensemble of trees by choosing different tree algorithms. We

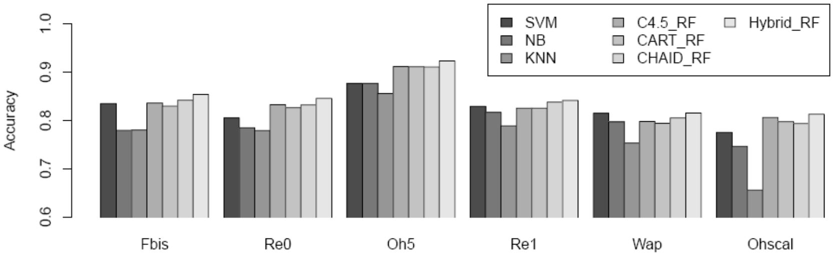


Fig. 3. Accuracy of the seven model builders

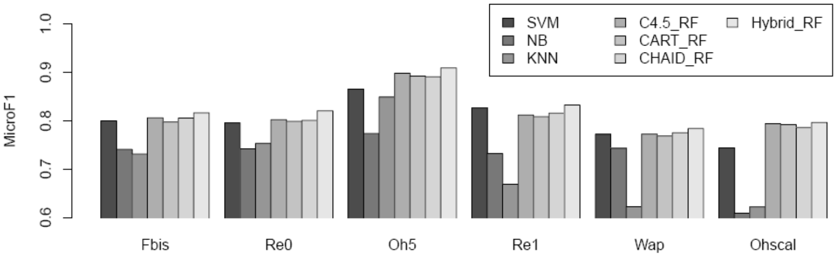


Fig. 4. MicroF1 for the seven model builders

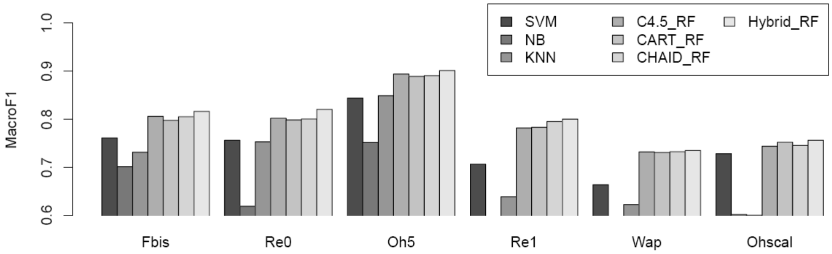


Fig. 5. MacroF1 for the seven model builders

demonstrate the advantage of our method in categorization. Our algorithm consistently improves classification performance. In the future work, we will consider alternative options for combining the three types of trees, rather than using the simple approach of keeping just one tree. For example, the results from the three trees might be combined into a single decision. Finally, alternative decision tree algorithms, or even other types of model builders, will be considered within this hybrid framework.

Acknowledgements. This research is supported in part by Shenzhen New Industry Development Fund under grant No.CXB201005250021A.

References

1. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
2. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)
3. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993)
4. Breiman, L., Friedman, J.H., Olshen R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth and Brooks/Cole Advanced Books and Software, Monterey, CA (1984)
5. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
6. Biggs, D., Suen, E.: A method of choosing multiway partitions for classification and decision trees. Journal of Applied Statistics 18(1), 49–62 (1991)
7. Ture, M., Kurt, I., Turhan Kurum, A., Ozdamar, K.: Comparing classification techniques for predicting essential hypertension. Expert Systems with Applications 29(3), 583–588 (2005)
8. Klema, J., Almonayyes, A.: Automatic categorization of fanatic texts using random forests. Kuwait Journal of Science and Engineering 33(2), 1–18 (2006)
9. Begum, N., Fattah, M.A., Ren, F.J.: Automatic text summarization using support vector machine. International Journal of Innovative Computing Information and Control 5(7), 1987–1996 (2009)
10. Chen, J.N., Huang, H.K., Tian, S.F., Qu, Y.L.: Feature selection for text classification with naive bayes. Expert Systems with Applications 36(3), 5432–5435 (2009)
11. Tan, S.: Neighbor-weighted K-nearest neighbor for unbalance text corpus. Expert Systems with Applications 28(4), 667–671 (2005)
12. Dietterich, T.G.: Machine learning research: Four current directions. AI Magazine 18(4), 97–136 (1997)
13. Yang, Y., Liu, X.: A re-examination of text categorization methods. In: ACM SIGIR 1999, pp. 42–49 (1999)
14. Han, E.-H(S.), Karypis, G.: Centroid-based Document Classification: Analysis and Experimental Results. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 424–431. Springer, Heidelberg (2000)
15. TREC. Text retrieval conference, http://trec.nist.gov
16. Lewis, D.D.: Reuters-21578 text categorization test collection distribution 1.0 (2011), http://www.research.att.com/~lewis
17. Hersh, W., Buckley, C., Leone, T.J., Hickam, D.: OHSUMED: An interactive retrieval evaluation and new large test collection for research. In: SIGIR 1994, pp. 192–201 (1994)
18. Moore, J., Han, E., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V., Mobasher, B.: Web page categorization and feature selection using association rule and principal component clustering. In: WITS 1997 (1997)
19. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In: AAAI Workshop 1998, pp. 41–48 (1998)
20. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann, Burlington (2011)

Learning Tree Structure of Label Dependency for Multi-label Learning*

Bin Fu[1], Zhihai Wang[1], Rong Pan[2], Guandong Xu[3], and Peter Dolog[2]

[1] School of Computer and Information Technology,
Beijing Jiaotong University, Beijing 100044, China
{09112072,zhhwang}@bjtu.edu.cn

[2] Department of Computer Science, Aalborg University, Denmark
{rpan,dolog}@cs.aau.dk

[3] School of Engineering & Science, Victoria University, Australia
Guandong.Xu@vu.edu.au

Abstract. There always exists some kind of label dependency in multi-label data. Learning and utilizing those dependencies could improve the learning performance further. Therefore, an approach for multi-label learning is proposed in this paper, which quantifies the dependencies of pairwise labels firstly, and then builds a tree structure of the labels to describe them. Thus the approach could find out potential strong label dependencies and produce more generalized dependent relationships. The experimental results have validated that compared with other state-of-the-art algorithms, the method is not only a competitive alternative, but also has shown better performance after ensemble learning especially.

Keywords: classification; multi-label instance; multi-label learning; label dependency.

1 Introduction

Classification is to predict possible labels on unlabeled instance given a set of labeled training instances. Traditionally, it is assumed that each instance is associated with only one label. However, an instance often has multiple labels simultaneously in practice [1,2]. For example, a report about *religion* could also be viewed as a *politics* report. Classification for this kind of instance is called multi-label learning. Nowadays, multi-label learning is receiving more and more concerns, and becoming an important topic.

Various methods have been developed for multi-label learning, and these methods mainly fall into two categories [2]: (1) algorithm adaptation, which extends traditional single-label models so that they can deal with multi-label instances directly. Several adapted traditional models include Bayesian method,

* This research has been partially supported by the Fundamental Research Funds for the Central Universities, China (2011YJS223), National Natural Science Fund of China (60673089, 60973011), and EU FP7 ICT project M-Eco: Medical Ecosystem Personalized Event-based Surveillance (No. 247829).

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 159–170, 2012.

AdaBoost, decision tree, associative rules, k-NN, etc. [3,4,5,6,7]. (2) problem transformation, which converts a multi-label problem into one or several single-label problems. Thus traditional single-label classifiers can be used directly without modification. Recently, many methods have been proposed to learn label dependency as a way of increasing learning performance [1,2,8,9,10,11,12]. However, most of them do not give an explicit description of label dependency. For example, classifier chain, a model proposed recently [9], links the labels into a chain randomly and assumes that each label is dependent on all its preceding labels in the chain. However, each label may be independent with its preceding labels while dependent on its following labels since they are linked randomly. Moreover, more complex dependency such as a tree or DAG-like hierarchical structure of labels often exists in practice, thus more appropriate models are needed to describe them.

Hence, we propose one kind of novel method for aforementioned issues. We quantify the dependencies of pairwise labels firstly, building a complete undirected graph that takes the labels as the set of vertices and the dependent values as edges. A tree is then derived to depict the dependency explicitly, so the unrelated labels can be removed for each label and the dependency model is generalized into a tree model. Furthermore, we also use ensemble technique to build multiple trees to capture the dependency more accurately. The experimental results would show our proposed method is competitive and could further enhance learning performance on most of datasets.

The remainder of this paper is organized as follows: We review the related works in section 2. A formal definition of multi-label learning is given in section 3. In section 4, we describe and analyze our proposed methods in detail. Section 5 is devoted to the experiment design and result analysis. The last section concludes this paper and gives some potential issues with further research.

2 Related Work

Many methods have been proposed to cope with multi-label learning by exploiting label's dependencies. According to the order of dependency be learned, these methods mainly fall into following categories.

(1) No label dependency is learned. Basic BR (Binary Relevance) method decomposed one multi-label problem into multiple independent binary classification problems, one for each label [2]. Boutell et al. used BR for scene classification [13]. Zhang et al. proposed ML-KNN, a lazy method based on BR [7]. Tsoumakas et al. proposed HOMER to deal with a large number of labels [14].

(2) Learning the dependencies of pairwise labels. Hullermeier et al. proposed the RPC method that learned the pairwise preferences and then ranked the labels [15]. Furnkranz et al. extended the RPC by introducing a virtual label [16]. Madjarov et al. proposed a two stage architecture to reduce the computational complexity of the pair-wise methods [17].

(3) Learning the dependencies within multiple labels. Basic LP (Label Powerset) method treated the whole set of labels as a new single label and learned dependencies within all them [2]. Tsoumakas et al. proposed the RAkEL$_d$

method that divided the label set into disjoint subsets of size k randomly [18]. Stacking-base method was proposed to aggregate binary predictions to form meta-instances [19]. Read proposed the PS, which decomposed the instance's labels until a threshold was met [8]. Read et al. proposed the CC (Classifier Chain) algorithm to link the labels into a chain randomly [9]. Dembcynski et al. proposed PCC, a probabilistic framework that solved the multi-label classification in terms of risk minimization [10].

A number of models have also been used to depict the labels dependencies explicitly, which include multi-dimensional Bayesian network, conditional dependency networks, conditional random fields [11,20,21,22,23]. Dembczynski et al. formally explained the difference between the conditional dependency and unconditional dependency [24]. Similar with these methods, our proposed method uses the tree to learn the label dependency explicitly. the difference is that we simply ignore the feature set in the process of constructing a tree, whereas others [11] build the models conditioned on the feature set.

3 The Concept of Multi-label Learning

Let X be the instance space, and $L = (l_1, l_2, \ldots, l_m)$ be a set of labels. Given a training instance set $D = \{(x_1, C_1), (x_2, C_2), \ldots, (x_n, C_n)\}$, where $x_k \in X$ is an instance, and $C_k \subset L$ is a subset of L denoting x_k's true labels, the target of multi-label learning is to build a classifier: $f : X \to 2^L$, that is a mapping from the instance space to a set of label subset, where 2^L is the power set of L. C_k can also be represented by a Boolean vector $(b_{k1}, b_{k2}, \ldots, b_{km})$, where $b_{kj} = 1$ indicates label l_j is x_k's true label ($l_j \in C_k$), while $b_{kj} = 0$ indicates the opposite.

Let $x \in X$ be an unlabeled instance, $y = (y_1, y_2, \ldots, y_m)$ be its Boolean vector of predicted labels, we can also make prediction by calculating the joint conditional probability$P(y|x)$. For each label l_k, let $Parent(l_k)$ denote the set of labels that label l_k is dependent on, $Parent(y_k)$ is the corresponding Boolean counterpart. Hence $P(y|x)$ can be transformed as Eq.(1).

$$P(y|x) = \prod_{k=1}^{m} P(y_k|parent(y_k), x) \tag{1}$$

where y_k denotes the kth label. Hence we can get the label vector's posterior probability by calculating each label's posterior probability respectively, so the transformation of Eq.(1) is a kind of problem transformation. A key issue is how to exactly find the set of dependent labels for each label in order to calculate the posterior probability more accurately.

4 Learning a Tree Structure of Labels

As mentioned above, to eliminate weak dependencies in CC model, and fit the real data more accurately, we propose a new algorithm named as LDTS (Learning dependency from Tree Structure of Labels) in this section.

LDTS firstly measures the dependency for each pairwise labels l_i and l_j, notated as $dependency(l_i, l_j)$, thus an undirected complete graph $G(L, E)$ is constructed, where the label set L denotes the vertices, and $E = \{dependency(l_i, l_j) : l_i \in L, l_j \in L\}$ denotes the edges. To determine the dependent labels for each label, a maximum spanning tree is then derived using Prim algorithm, and each label is assumed to be dependent on its ancestor labels. A dataset is then created for each label and their dependent labels are added into the feature set, so we could utilize these dependency since the classifiers is trained based on the new feature set. The whole training process is outlined in Algorithm 1.

Algorithm 1. The process of training LDTS classifier

```
Input:
    The training dataset: $D = \{(x_1, C_1), (x_2, C_2), \ldots, (x_n, C_n)\}$;
    The algorithm for training base classifier: $B$.
Output:
    Classifiers: $(f_1, f_2, \ldots, f_m)$.
 1: for each pair of labels $(l_i, l_j)$, measure their dependency: $dependency(l_i, l_j)$
 2: create a undirected full graph $G = (L, E)$
 3: use the $Prim$ algorithm to derive a maximum spanning tree: $T$
 4: for label $l_1, l_2, ..., l_m$, get the set consists of its ancestor labels: $Parent(l_i)$
 5: for $i = 0$ to $m$ do
 6:     let the $D_i = D$
 7:     for $j = 0$ to $m$ do
 8:         if $l_j \notin Parent(l_i)$ then
 9:             delete this label from $D_i$
10:         end if
11:     end for
12:     for the $D_i$, set $l_i$ as its only label, train the classifier $f_i$
13: end for
14: return the $m$ classifiers: $(f_1, f_2, \ldots, f_m)$
```

At step 1, mutual information is used to compute the dependencies of pairwise labels. Its definition is shown as follows [25].

$$H(X, Y) = \sum_{x,y} P(x, y) \log \frac{P(x, y)}{P(x)P(y)} \tag{2}$$

where X and Y are two variables, x and y are their all possible values.

The labels is organized using a tree for two purposes. Firstly, the properties of maximum spanning tree ensure that each label is more dependent on its ancestor labels than other labels, since they have greater mutual information value. Hence it could eliminate weak dependency further by assuming each label is only dependent on its ancestor labels. When generating the graph and tree of labels, we simply assume that label dependency is independent with the feature set and only consider the mutual influence among the labels, this is one kind of the unconditional dependency described in [24]. Secondly, various kinds of dependencies including tree hierarchy and DAG of dependency may exist within labels. Therefore, we expect the performance could be improved, especially on the datasets in which the labels are organized into a tree indeed.

It is also should be noted that the main purpose of our method is to find more accurate label dependency, and it does not impose a hierarchy on labels since labels have the same parent or in different paths could exist simultaneously. This is different from the hierarchical classification that impose a strict label hierarchy. Although the randomness in not eliminated fully, we do reduce it to only select the strong dependency randomly. One possible issue is that a full graph of labels needs to be learned with the computational complexity $O(n^2)$, the efficiency needs further improvement to cater for a large number of labels.

When classifying an unlabeled instance x, each label l_i can not be predicted until its dependent labels are all predicted. Hence the labels should be predicted from the root of the tree and then its children recursively until all the leaves are reached. The detailed process is depicted in Algorithm 2. For each label, the labels dependencies are considered since its prediction is based on the feature set and the predictions of its dependent labels.

Algorithm 2. The process of classification using LDTS classifier

```
Input:
    A unlabeled instance $x$ that needs to be classified.
Output:
    The prediction $Y = (y_1, y_2, \ldots, y_m)$.
 1: set the vector to be empty, $Y \leftarrow ()$
 2: set the root label as the current label need to be predicted: $t$
 3: predict current label $t$ for $x$ using corresponding classifier $f_t$
 4: add $f_t(x)$ into $Y$, $Y \leftarrow Y \bigcup f_t(x)$
 5: use the result $f_t(x)$ to update the $x$, $x = (x, f_t(x))$
 6: find all the children labels of $t$
 7: repeat
 8:    for each children labels $c_i$ of $t$ do
 9:       repeat the step 2-6
10:    end for
11: until all the labels are predicted
12: return the predicted vector $Y$
```

When generating the directed tree in LDTS, the root node is selected randomly. However, selecting a different label will result in a different tree and thus generating different dependent labels for each label. Another issue is the label dependency could not be utilized fully, since the dependency of pairwise labels l_i, l_j calculated here is mutual and useful equally to each other. One possibility is that a label may also depend on its children labels, but the directed tree does not allow for this situation. To address such issues, the ensemble learning is used to generate multiple LDTS classifiers iteratively. In each iteration, the classifier is trained on a sampling of the original dataset, and the root label is reselected randomly. Hence each iteration will get a different label tree and combining them will reduce the influence of the root's randomness and take full advantage of the label dependency. We call this extended method ELDTS(Ensemble of LDTS).

The detail process is depicted in Algorithm 3. Given an unlabeled instance x, all predictions of these classifiers will be aggregated into a final result by voting simply.

Algorithm 3. The process of training ELDTS classifier

```
Input:
    The training dataset: $D = \{(x_1, C_1), (x_1, C_1), \ldots, (x_n, C_n)\}$;
    The algorithm for training base classifier: $B$;
    The number of iteration: $n$.
Output:
    An ensemble of LDTS classifiers $F = (f_1, f_2, \ldots, f_m)$.
set the $F$ to be empty: $F = ()$
for $i = 0$ to $m$ do
    generate a new dataset $D_i$ by sampling on the original dataset with replacement

    select the root label $r_i$ randomly;
    train a LDTS classifier $t_i$ using B, based on the dataset $D_i$ and root label $r_i$
    add $f_i$ into $F$
end for
return the ensemble of classifiers: $F$
```

All above are the description and analysis of our proposed algorithms. Comparison with other state-of-the-art algorithms and further analysis will be given in the following section.

5 Experiment Design and Analysis

5.1 The Description of Datasets

We take several datasets from multiple domains for the experiments, and table 1 depicts them in detail.

Table 1. Description of the datasets used in experiments

Dataset	Domain	Instances	Attributes	Labels	LC	LD	DLS
emotions	music	593	72	6	1.869	0.311	27
enron	text	1702	1001	53	3.378	0.064	753
medical	text	978	1449	45	1.245	0.028	94
scene	image	2407	294	6	1.074	0.179	15
yeast	biology	2417	103	14	4.237	0.303	198

Several statistics as follows have been used to characterize these datasets.

(1) Label cardinality: LC $= \frac{1}{n}\sum_{i=1}^{n} |C_i|$. It calculates the average number of labels for each instance, where $|C_i|$ is the number of true labels of the ith instance.

(2) Label density: LD $= \frac{1}{n}\sum_{i=1}^{n} \left|\frac{C_i}{m}\right|$. It is calculated by dividing the label cardinality by m, the size of original label set.

(3) Distinct label sets: $\text{DLS}(D) = |\{C|\exists(x, C) \in D\}|$. It counts the number of distinct label sets that appear in the dataset.

Seen from table 1, these datasets cover many domains including text categorization, scene classification, emotion analysis, biology etc.. It should be noted that there are no label hierarchies in these datasets and we use them to examine whether our methods could find more strong label dependency and gain better performance. More detail description can be found on the official website of Mulan[1].

5.2 Evaluation Criteria

In order to evaluate the algorithm performance, the criteria should be specified. Let $D = \{(x_1, C_1), (x_2, C_2), \ldots, (x_n, C_n)\}$ be a dataset, where x_i is the ith instance, and $C_i \subset L$ is its true labels. Given a classifier f and an instance x_i, Y_i denotes the predicted labels for x_i, while $rank(x_i)$ or $rank_i$ denotes the predicted rank of labels, and $rank(x_i, l)$ denotes the label l's position in the rank. All the criteria we used are as follows.

(1) Hamming loss: It is proposed by Schapire and Singer [4].

$$\text{H-Loss}(f, D) = \frac{1}{n} \sum_{i=1}^{n} \frac{Y_i \bigoplus C_i}{m} \tag{3}$$

The operator $\bigoplus$ calculates the symmetric difference of two sets, which is the number of misclassified labels for an instance.

(2) Accuracy: It calculates the ratio between the intersection and union of the predicted set of labels and the true set of labels for the instances on average.

$$\text{Accuracy}(f, D) = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{Y_i \bigcap C_i}{Y_i \bigcup C_i} \right| \tag{4}$$

(3) One-error: It calculates how many times that top-ranked label is not a true label of the instance.

$$\text{One-Error} = \frac{1}{n} \sum_{i=1}^{n} \delta(\arg \min_{l \in L} rank(x_i, l)) \tag{5}$$

where $\delta(x) = 1$ if l is a true label of the instance, otherwise $\delta(x) = 0$.

(4) Ranking loss: It expresses the number of times when the irrelevant labels are ranked before the true labels.

$$\text{R-Loss} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\left|\overline{C_i}\right| |C_i|} \left|\{(l_a, l_b) : rank(x_i, l_a) > rank(x_i, l_b), (l_a, l_b) \in C_i \times \overline{C_i}\}\right| \tag{6}$$

[1] http://mulan.sourceforge.net/

(5) Average precision: This measurement calculates the average fraction of labels ranked above a particular label $l \in C_i$, which are all also in C_i.

$$\text{AvePrec} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{|\overline{C_i}|} \sum_{l \in C_i} \frac{|\{l' \in C_i : rank(x_i, l') \leq rank(x_i, l)\}|}{rank(x_i, l)} \tag{7}$$

These criteria evaluate the different aspects of these methods. While Hamming loss and accuracy do not consider the relation between different predictions, the other 3 criteria take such a relation into considerations, since they are based on the ranking of the probabilities predicted for all labels. Because our methods are intended to get a more accurate probability for each label by finding more strong labels dependencies, thus for each label, it should be predicted more accurately and the true labels should be given greater possibilities. Therefore, we expect that our method could gain better performance under Hamming loss and ranking loss, since Hamming loss examines the predictions of all labels independently and ranking loss focuses on whether the true labels are given greater probabilities than other labels. For other 3 criteria, our method may be effective, but they are not what our method optimize for.

5.3 Algorithms and Settings

The algorithms used for comparison are listed in table 2 with their abbreviations respectively. To examine the effect of label dependency, BR algorithm is used as a baseline since it does not consider the label dependency, then we compare our proposed LDTS and ELDTS with CC and ECC methods to see their effectiveness after eliminating weak dependencies. RAkEL$_d$ and RAkEL are also used for comparison as other ways of learning label dependency.

The experiments are divided into two parts, according to the two purposes mentioned in section 4. One part is on the five aforementioned datasets without label hierarchy to see whether our method can find more strong dependency and thus gain better performance, the other part is on the dataset rcv1v2, a dataset in which there exists a tree hierarchy of labels, to see its performance when a tree structure is learned. Since only one tree exists in rcv1v2, we do not use the ensemble method on it.

Table 2. The algorithms used for comparison

Compared with LDTS	Compared with ELDTS
BR: Binary relevance method	EBR: Ensemble of BR
CC: Classifier chain	ECC: Ensemble of CC
RAkEL$_d$: Random disjoint k label subsets	RAkEL: Random k label subsets

All algorithms are implemented on the Mulan framework [26], an open platform for multi-label learning. The parameter values are chosen as those used in the paper [9]. For the RAkEL, we set the $k = \frac{m}{2}$. For the RAkEL$_d$, we set the

$k = 3$. For the ensemble algorithms, the number of iterations is 10, and for each iteration, 67% of the original dataset is sampled with replacement to form the training dataset. SMO, a support vector machine classifier implemented in Weka [27], is used as the base classifier. All algorithms are executed 5 times using 10-fold cross validation on all datasets expect rcv1v2 with different random seeds 1, 3, 5, 7, 11, respectively, and the final results are the averaged values. For the rcv1v2, only 100 attributes are kept, and 10-fold cross validation is used only one time since it has a huge amount of instances and attributes.

5.4 Experimental Results and Analysis

Based on above setup, we get the final results and the following tables display them in detail. The bold result indicates the best one, and result with the black dot "●" indicates our proposed algorithm is better than itself indicated algorithm.

Table 3. The hamming loss of each algorithm on the datasets

Dataset	BM	CC	LDTS	EBM	ECC	ELDTS	RAkEL
emotions	**0.1939**	0.2159●	0.2038	0.1947●	0.2108●	**0.1932**	0.2283●
enron	**0.0601**	0.0606●	0.0602	0.0540●	0.0536●	**0.0535**	0.0541●
medical	0.0101●	**0.0098**	0.0099	0.0098●	0.0096●	**0.0095**	0.0102●
scene	0.1046	**0.1037**	0.1056	0.1020●	0.0997●	**0.0968**	0.1047●
yeast	**0.1990**	0.2115●	0.2060	**0.1991**	0.2106●	0.2034	0.2371●

Table 4. The accuracy of each algorithm on the datasets

Dataset	BM	CC	LDTS	EBM	ECC	ELDTS	RAkEL
emotions	0.5199●	0.5336●	**0.5727**	0.5226●	0.5451●	**0.5808**	0.5119●
enron	0.4058●	0.4083●	**0.4085**	0.4370●	0.4110●	**0.4396**	0.4063●
medical	0.7580●	**0.7750**	0.7670	0.7655●	**0.7799**	0.7759	0.7686●
scene	0.5999●	**0.6949**	0.6496	0.6137●	**0.7020**	0.6783	0.6281●
yeast	0.5003●	0.4879●	**0.5073**	0.5031●	0.4929●	**0.5249**	0.4692●

As shown from table 3 to table 7, our proposed LDTS method performs better on the majority of datasets evaluated by the criteria. It is superior to CC on 3 datasets under the Hamming loss, accuracy, one-error, and ranking loss, but inferior under other metrics. LDTS algorithm does not improve all the time or the improvement is not significant. The possible reason is that although LDTS algorithm could learn the dependency further, it still ignore lots of useful dependency since it only considers unidirectional dependency of pairwise labels, especially when the labels are dependent mutually. We expect that ensemble learning that combines different trees can further utilize the label dependency , since the dependent direction between pairwise labels is changed in a different tree by choosing a different root.

To validate above assumption, we also use ensemble learning on these algorithms and compare them each other. Also shown from table 3 to table 7, our proposed ELDTS has a substantial improvement after the employing ensemble learning. Under all 5 criteria, ELDTS is superior to ECC on most datasets. These results show that through learning multiple label trees by ensemble learning, the influence of the root label's randomness could be mitigated, and the label dependencies are learned more effectively. Although ECC algorithm also changed the order of labels, it could not make sure that only the strong dependency is considered each time, since it's totally random when determining the dependent relationship within labels.

It can be observed that the proposed algorithms do not perform well on two datasets *medical* and *scene*. Seen from the table 1, these two datasets have very small label cardinality, which means there tend to be less label dependency in them and overemphasis on label dependency may not be preferable. Thus the algorithms we propose are more suitable for the datasets that there are indeed strong label dependency in them.

The results gotten on rcv1v2, a dataset with tree structure of labels, are also given in table 8. We can clearly see that our proposed LDTS method is superior under Hamming loss and ranking loss, the criteria it optimize for. Therefore, it has been proven that our method is more effective when there is complex dependency within labels.

Table 5. The one-error of each algorithm on the datasets

Dataset	BM	CC	LDTS	EBM	ECC	ELDTS	RA*k*EL
emotions	**0.2989**	0.3700●	0.3103	**0.2534**	0.3181●	0.2563	0.3096●
enron	0.4912●	0.4938●	**0.4897**	0.3078●	0.3071●	**0.3054**	0.3210●
medical	0.2029●	**0.1847**	0.1932	0.1407●	0.1415●	**0.1397**	0.1634●
scene	0.3389●	**0.2793**	0.3132	0.2553●	0.2609●	**0.2485**	0.2777●
yeast	**0.2557**	0.2559●	0.2557	0.2557●	0.2583●	**0.2551**	0.2895●

Table 6. The rank loss of each algorithm on the datasets

Dataset	BM	CC	LDTS	EBM	ECC	ELDTS	RA*k*EL
emotions	0.2778●	0.2876●	**0.2334**	0.2169●	0.2317●	**0.1743**	0.1964●
enron	**0.2915**	0.2929●	0.2925	0.1648●	0.1640●	**0.1622**	0.1784●
medical	0.0952●	**0.0926**	0.0932	0.0537●	**0.0516**	0.0518	0.0598●
scene	0.1718●	**0.1576**	0.1664	0.1162●	0.1115●	**0.0945**	0.1110●
yeast	0.3188●	0.3351●	**0.3181**	0.2739●	0.2771●	**0.2273**	0.2300●

Table 7. The average precision of each algorithm on the datasets

Dataset	BM	CC	LDTS	EBM	ECC	ELDTS	RA*k*EL
emotions	0.7384●	0.7159●	**0.7634**	0.7814●	0.7573●	**0.8040**	0.7734●
enron	0.4682●	**0.4702**	0.4693	0.6284●	0.6287●	**0.6314**	0.6011●
medical	0.7977●	**0.8115**	0.8066	0.8672●	**0.8710**	0.8707	0.8524●
scene	0.7752●	**0.8048**	0.7881	0.8353●	0.8351●	**0.8479**	0.8285●
yeast	0.6697●	0.6611●	**0.6728**	0.7003●	0.6975●	**0.7253**	0.7048●

Table 8. The performance of algorithms on dataset rcv1v2

criterion	BM	CC	LDTS	RAkEL$_d$
H-Loss	**0.0235**	0.0294●	**0.0235**	0.0237●
Accuracy	0.1810●	**0.2529**	0.2237	0.1783●
One-Error	0.7638●	**0.7065**	0.7120	0.9538●
R-Loss	0.3560●	0.3508●	**0.3381**	0.4590●
AvePrec	0.2537●	**0.3085**	0.2940	0.1001●

6 Conclusion

In this paper, one kind of novel approaches are proposed to exploit the label dependency. Specifically, the dependency degree of pairwise labels is calculated firstly and then a tree is build to represent the dependency structure of labels. The methods assume that the dependencies only exist between each label and its ancestor labels, resulting in reducing the influence of weak dependency. At the same time, they also generalize the label dependency into a tree model. Furthermore, we utilize ensemble learning to learn and aggregate multiple label trees to reflect the labels dependencies fully. The experimental results show that the algorithms we proposed perform better, especially after boosted by the ensemble learning.

One potential problem is that using mutual information to measure the dependency will give equal values to both of the labels, which assumes that the dependency for pairwise labels is mutual and equal for each other. However, the label dependency could be directed possibly and this assumption is often violated in reality. Hence how to measure the directed label dependency should be one of the next directions. Additionally, how to generalize the tree structure of labels further to graph or forest structure is another issue in the future work.

References

1. Cheng, W., Hullermeier, E.: Combining Instance-Based Learning and Logistic Regression for Multilabel Classification. Machine Learning 76(2-3), 211–225 (2009)
2. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining Multi-label Data. In: Oded, M., Lior, R. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 667–685. Springer, New York (2010)
3. McCallum, A.K.: Multi-label Text Classification with a Mixture Model Trained by EM. In: Proceedings of AAAI 1999 Workshop on Text Learning (1999)
4. Schapire, R.E., Singer, Y.: Boostexter: a Boosting-Based System for Text Categorization. Machine Learning 39(2-3), 135–168 (2000)
5. Clare, A.J., King, R.D.: Knowledge Discovery in Multi-label Phenotype Data. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, pp. 42–53. Springer, Heidelberg (2001)
6. Thabtah, F.A., Cowling, P., Peng, Y.: MMAC: a New Multi-class, Multi-label Associative Classification Approach. In: Proceedings of the 4th International Conference on Data Mining, pp. 217–224 (2004)
7. Zhang, M., Zhou, Z.: ML-KNN: A Lazy Learning Approach to Multi-label Learning. Pattern Recognition 7(40), 2038–2048 (2007)
8. Read, J.: Multi-label Classification using Ensembles of Pruned Sets. In: Proceedings of the IEEE International Conference on Data Mining, pp. 995–1000. IEEE Computer Society, Washington, DC (2008)

9. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier Chains for Multi-label Classification. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009, Part II. LNCS, vol. 5782, pp. 254–269. Springer, Heidelberg (2009)
10. Dembczynski, K., Cheng, W., Hullermeier, E.: Bayes Optimal Multilabel Classification via Probabilistic Classifier Chains. In: Proceedings of the 27th International Conference on Machine Learning, pp. 279–286. Omnipress (2010)
11. Zhang, M., Zhang, K.: Multi-label Learning by Exploiting Label Dependency. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 999–1000. ACM Press, Washington, DC (2010)
12. Zhang, Y., Zhou, Z.: Multi-label Dimensionality Reduction via Dependence Maximization. ACM Transactions on Knowledge Discovery from Data 4(3), 1–21 (2010)
13. Boutell, M.R., Luo, J., Shen, X.: Learning Multi-label Scene Classification. Pattern Recognition 37(9), 1757–1771 (2004)
14. Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective and Efficient Multilabel Classification in Domains with Large Number of Labels. In: Proceedings of ECML/PKDD 2008 Workshop on Mining Multidimensional Data, pp. 30–44 (2008)
15. Hullermeier, E., Furnkranz, J., Cheng, W.: Label Ranking by Learning Pairwise Preferences. Artificial Intelligence 172(16-17), 1897–1916 (2008)
16. Furnkranz, J., Hullermeier, E., Mencia, E.L.: Multilabel Classification via Calibrated Label Ranking. Machine Learning 2(73), 133–153 (2008)
17. Madjarov, G., Gjorgjevikj, D., Dzeroski, S.: Two Stage Architecture for Multi-label learning. Pattern Recognition 45(3), 1019–1034 (2011)
18. Tsoumakas, G., Katakis, I., Vlahavas, I.: Random k-labelsets for Multi-label Classification. IEEE Transactions On Knowledge and Data Engineering 23(7), 1079–1089 (2011)
19. Tsoumakas, G., Dimou, A., Spyromitros, E.: Correlation-Based Pruning of Stacked Binary Relevance Models for Multi-Label Learning. In: Proceeding of ECML/PKDD 2009 Workshop on Learning from Multi-Label Data, Bled, Slovenia, pp. 101–116 (2009)
20. Gaag, L., Waal, P.: Multi-dimensional Bayesian Network Classifiers. In: Third European Workshop on Probabilistic Graphical Models, pp. 107–114 (2006)
21. Bielza, C., Li, G., Larranage, P.: Multi-dimensional Classification with Bayesian Networks. International Journal of Approximate Reasoning 52(6), 705–727 (2011)
22. Guo, Y., Gu, S.: Multi-label Classification using Conditional Dependency Networks. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, pp. 1300–1305 (2011)
23. Ghamrawi, N., McCallum, A.K.: Collective Multi-label Classification. In: Proceedings of the 2005 ACM Conference on Information and Knowledge Management, pp. 195–200 (2005)
24. Dembczynski, K., Waegeman, W., Cheng, W.: On Label Dependence in Multi-label Classification. In: Proceedings of the 2nd International Workshop on Learning From Multi-label Data, pp. 5–12 (2010)
25. Chow, C.K., Liu, C.N.: Approximating Discrete Probability Distributions with Dependency Trees. IEEE Transactions on Information Theory 14(3), 462–467 (1968)
26. Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I.: Mulan: A Java Library for Multi-Label Learning. Journal of Machine Learning Research 12, 2411–2414 (2011)
27. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Francisco (2000)

Multiple Instance Learning for Group Record Linkage

Zhichun Fu[1], Jun Zhou[1], Peter Christen[1], and Mac Boot[2]

[1] Research School of Computer Science
College of Engineering and Computer Science
The Australian National University
Canberra ACT 0200, Australia
{sally.fu,jun.zhou,peter.christen}@anu.edu.au

[2] Australian Demographic and Social Research Institute
College of Arts and Social Sciences
The Australian National University
Canberra ACT 0200, Australia
mac.boot@anu.edu.au

Abstract. Record linkage is the process of identifying records that refer to the same entities from different data sources. While most research efforts are concerned with linking individual records, new approaches have recently been proposed to link groups of records across databases. Group record linkage aims to determine if two groups of records in two databases refer to the same entity or not. One application where group record linkage is of high importance is the linking of census data that contain household information across time. In this paper we propose a novel method to group record linkage based on multiple instance learning. Our method treats group links as bags and individual record links as instances. We extend multiple instance learning from bag to instance classification to reconstruct bags from candidate instances. The classified bag and instance samples lead to a significant reduction in multiple group links, thereby improving the overall quality of linked data. We evaluate our method with both synthetic data and real historical census data.

Keywords: Multiple instance learning, record linkage, entity resolution, instance classification, historical census data.

1 Introduction

Within many organisations, data are collected from various sources and through different channels, and they are stored in databases with different structures and formats. As organisations collaborate, data often need to be exchanged and integrated. The objective of such data integration is to identify and match all records that correspond to the same real-world entity, such as the same customer, patient, or taxpayer [10]. Record linkage (also known as data matching or entity resolution) is a key step to effectively mine rich information that is not available in a single database. This technology has been used in many areas, such as

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 171–182, 2012.

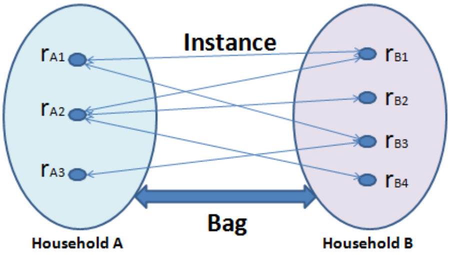


Fig. 1. An example of group (household) record linkage, and the corresponding MIL setting. Links between individual records correspond to instances while a bag is made of all links between the records in two groups.

electronic health record systems, the retail industry, business analytics, fraud detection, demographic tracking, and government administration [10].

As one application of record linkage, linking of historical census records across time can greatly enhanced their values by, for example, enabling tracking of households and providing new insights into the dynamic character of social, economic and demographic changes. In recent years, researchers have tried to link records between census datasets using automatic or semi-automatic methods [3,15,19,21]. Unfortunately, these attempts have not been not very successful in linking records that correspond to individuals in a household [20].

Several reasons make the linking of historical census records a challenging undertaking. First, the quality of historical census data is poor, because large amounts of errors and inaccurate information have been introduced during the census collection and digitisation processes [19]. Second, a large portion of records contain the same or similar values. It is not uncommon to find different people with the same name, the same age, and living in the same street in one dataset. Third, the structure of households and their members can change significantly between two censuses (which were normally collected every five or ten years). Therefore, simply comparing individual records does not lead to reliable linkage outcomes. Considering household information in the linkage process can help overcome this challenge.

In this research, we tackle the problem of linking individual records and households in historical census data. A household link will likely contain several links between individual record pairs for its household members. If two households are matching, at least one of their record links has to be a match. On the contrary, if two households are not matching, none of their record links shall be matched. This is a typical multiple instance learning (MIL) setting. MIL is a supervised learning method proposed by Dietterich et al. [9]. In MIL, data are represented as bags, each of which contains some instances. In a binary classification setting, a positive bag contains both positive and negative instances, while a negative bag only consists of negative instances. In the training stage, the class labels are only available at the bag-level but not at the instance-level. The goal of MIL is to learn a classifier which can predict the label of an unseen bag. When applying

MIL to the group record linkage problem, group links are treated as bags, and record links become the instances in these bags. A model can then be learned to classify a group link as a match or non-match. Figure 1 shows an example of group linking and its relationship to the MIL setting.

Because an individual record in one census dataset has generally a high similarity with several records in different households in another dataset, a household in one census dataset is often linked to different households in another dataset. Although such results can be helpful, e.g. in generating family trees, social scientists are often interested in tracking the majority of household members as a whole entity over time [20]. This suggests one-to-one household matches are needed. To reduce the number of multiple household matches, we can employ a group linking method [17,18], which generates a household match score for each household pair. Then the household pairs with the highest match score are selected as the final match results. Such an approach requires the detection of all matched record pairs in a household, which is equivalent to classifying instances within a bag as matches or non-matches. This is a problem that has not been adequately addressed in MIL research [16]. In traditional MIL methods [4,14], when instance selection is concerned, only the optimal positive instances are explored, whilst no explicit instance classification solution has been given. Therefore, there is a gap between MIL and its application to group record linkage.

We extend the above mentioned MIL methods to instance level classification by grouping negative instances from the training set with an instance to be classified. This transforms the instance into a bag. We can then employ the bag-level classification model for explicit instance classification. We show that this method can effectively classify both household and record links.

This paper makes two contributions. First, we extend the MIL method to instance classification via bag reconstruction. Second, we propose a practical solution to linking households between historical census datasets by group linkage using MIL. Our method is general in nature and it can be applied to other record linkage applications that require groups of records rather than individual records to be linked.

2 Related Work

In recent years, many methods have been developed for record linkage in the fields of machine learning, data mining and database systems [10]. Among them, supervised learning has been intensively investigated. It uses labelled record pairs with known match status (match or non-match) to learn a classification model. Bilenko et al. [2] proposed a solution based on support vector machines (SVM) [23] to compute the similarity between strings. Alternatively, Christen [6] has constructed inputs for a SVM using a pre-selection step which retrieves record pairs that with high confidence correspond to matches or non-matches. These pairs then become the positive and negative training samples for a SVM classifier. This method can be considered as a combination of supervised and un-supervised techniques.

Group record linkage methods have been developed to process groups rather than individual records [18]. On et al. [17] defined group similarity from two aspects, the similarity between matched record pairs and the fraction of matched record pairs between two groups of records. A group similarity can then be calculated using a maximum weight bipartite matching.

Multiple instance learning is a paradigm of machine learning that deals with a collection of data called *bags*. The original work by Dietterich et al. [9] attempted to recover an optimal axis-parallel hyper-rectangle in the instance feature space to separate instances in positive bags from those in negative bags. Departing from this model, several researchers have extended the framework, such as MI-SVM [1], DD-SVM [5], SMILE [24], MILES [4] and MILIS [14].

Among these works, we are particularly interested in the Multiple Instance Learning with Instance Selection (MILIS) method because it allows efficient and effective instance prototype selection for target concept representation [14]. This is an important property for (historical) census record linkage, which works on potentially large numbers of households and their records, and contains significant amounts of uncertainty because of low data quality.

MILIS is an extension of MIL using an embedded instance selection (MILES) method [4]. The general idea of these two methods is to map each bag into a feature space defined by selected instances, which is based on bag-to-instance similarity. It generates a feature vector for each bag, whose dimension is the number of selected instances. In this manner, the MIL problem is converted into a supervised learning problem, for which a SVM can be used for classification.

The major difference between MILES and MILIS methods is on the instance selection step. In MILES, all instances in the training set are used for feature mapping, then important features are selected by a 1-norm SVM. Because the total number of instances in a training set may be very large, MILES can be very time consuming. MILIS, however, only selects one instance prototype (IP) from each bag for the embedding. It generates a feature space with much smaller dimension than MILES. The selection of IPs is done through a two-step optimisation framework, which updates IPs and a SVM classifier iteratively.

3 Group Linkage Using Multiple Instance Learning

In this section, we introduce a group record linkage method based on multiple instance learning. Here, we treat a group link as a bag and its record links as instances in a bag, as shown in Figure 1. As mentioned before, group linking requires prediction on whether or not two records match, which is equivalent to instance classification. Therefore, we extend the MILIS algorithm so that a single instance can be grouped with negative instances in the training set to create a new bag. Then the bag can be classified using the learned bag-level classifier.

3.1 Instance Selection and Classifier Learning

To commence, we give formal definitions of the notion used in the method. Let $\mathcal{B}^+ = \{B_1^+, \ldots, B_{n^+}^+\}$ be a set of positive bags, $\mathcal{B}^- = \{B_1^-, \ldots, B_{n^-}^-\}$ be a set of negative bags, and $n = n^+ + n^-$ be the total number of bags in the training set. A bag B_i contains m_i instances denoted by $\mathbf{x}_{i,j}$ for $j = 1, \ldots, m_i$, with the value for m_i varying from bag to bag. Each instance $\mathbf{x}_{i,j}$ is associated with a label $y_{i,j} \in \{1, -1\}$ that is not directly observable in the MIL setting, with $y_{i,j} = 1$ corresponding to a match and $y_{i,j} = -1$ to a non-match. The purpose is, therefore, to predict the binary label value $y_i \in \{1, -1\}$ for a novel test bag $B_i = \{\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,k}\}$, and $y_{i,j}$ for an instance $\mathbf{x}_{i,j}$.

Following the idea of instance-based embedding in [4] and instance prototype selection in [14], we generate bag-level feature representation using the similarity between a bag and an instance

$$s(B_i, \mathbf{x}) = \max_{\mathbf{x}_{i,j} \in B_i} \exp\left(-\gamma ||\mathbf{x}_{i,j} - \mathbf{x}||^2\right), \tag{1}$$

where γ is a feature mapping parameter that controls the similarity. Then a bag can be represented as an n-dimensional vector

$$z_i = [s(B_i, \mathbf{x}_1^*), \ldots, s(B_i, \mathbf{x}_i^*), \ldots, s(B_i, \mathbf{x}_n^*)], \tag{2}$$

where $\mathbf{x}_i^*$ are the prototype instances selected from the training set.

As proposed in [14], instance prototypes can be generated by selecting the least negative instance from each positive bag and the most negative instance from the negative bag. This requires modelling of the distribution of negative instances, and computing the probability that an instance has been generated from the negative population. Given an instance $\mathbf{x}$ and its k-nearest negative instances from the negative bags X_k^-, the likelihood of $\mathbf{x}$ being negative is

$$p(\mathbf{x}|X^-) = \frac{1}{Z} \sum_{j=1}^{k} \exp\left(-\beta ||\mathbf{x} - \mathbf{x}_j^-||\right), \tag{3}$$

where $\mathbf{x}_j^- \in X^-$ is the j^{th} nearest negative neighbour of $\mathbf{x}$, Z is a normalisation factor, and β is a parameter to control the contribution from training samples. We then select the instance with the lowest likelihood value from each positive bag as the positive instance prototypes (PIPs), and the instance with the highest likelihood value from each negative bag as negative instance prototypes (NIPs). These PIPs and NIPs form the set of instance prototypes (IPs) used in the feature mapping. Using Equations 2 and 3, we can represent bags in the training set in vector form, and then train a SVM classifier by solving the following unconstrained optimisation problem:

$$\min_{\mathbf{w}} \frac{||\mathbf{w}||^2}{2} + C \sum_i \max(1 - y_i(\mathbf{w}^T \mathbf{z}_i), 0), \tag{4}$$

where $y_i \in \{1, -1\}$ is the label for bag i, $\mathbf{w}$ is a set of parameters that define a separating hyper-plane, and C is the regularisation parameter [23].

3.2 Instance Classification

Both MILES and MILIS can find the most positive instance in a positive bag. This is achieved by selecting an instance in the bag that has the lowest likelihood value using Equation 3, because a positive bag should contain at least one positive instance. However, when it comes to the situation where a bag contains more than one positive instance, neither method provides an explicit solution to finding all the positive instances. Although a threshold may be set for decision, with instances whose likelihood is higher than the threshold classified as positive, and visa versa, it is practically difficult to find an appropriate threshold.

Here we propose a method for instance classification by bag reconstruction. We treat each instance in a positive bag as a seed, and group the instance with negative instances to create new bags. Then we apply the trained bag-level classifier to these new bags. If a new bag is classified as positive, then the seed instance is classified as positive. Otherwise, it is classified as negative. This method is based on the fact that if a seed is negative, the reconstructed bag consists of negative instances only, and thus will be classified as negative. Otherwise, the new bag contains one positive instance, therefore, is very likely to be classified as positive.

We have adopted two strategies for the bag reconstruction, **Random** and **Greedy**, to cope with multiple positive instances in a candidate bag. The first strategy randomly selects negative instances from the training set and groups them with the seed. Therefore, both the random negative instances from the training set and the seed instance contribute to the embedding step in MIL. The second strategy is built on top of the random option. With randomly selected negative instances, a greedy algorithm is adopted which reconstructs new bags and predicts the label of the newly added instance simultaneously. This guarantees not only the seed, but also the negative instances in the candidate bag, contribute to the embedding step. For each instance $\mathbf{x}$ in the candidate bag, we compute its Hausdorff distance to a bag G that contains NIPs $\mathbf{x}_i^{*-}$ only:

$$d(G, x) = \min_{\mathbf{x}_i^{*-} \in G} ||\mathbf{x} - \mathbf{x}_i^{*-}||^2 \tag{5}$$

Using this distance measure, we can get the similarity between an instance and the negative instances in G. By ranking the distances, we can construct a new bag by sequentially adding into the bag an instance with the lowest distance among the rest of the instances in the candidate bag. Evaluating the new bag using the bag-level SVM classifier, we can get the label of the newly added instance. For a candidate bag that contains both positive and negative instances, initially, the added instances are negative. Therefore, the bag is predicted as negative. When the prediction becomes positive after a new instance is added, the new instance is classified as positive. We then replace the positive instance with an instance that has a larger distance, and re-evaluate the new bag. This process continues until all instances in the candidate bag have been traversed. We summarise this strategy in Algorithm 1.

Algorithm 1. Instance Classification using Greedy Bag Reconstruction

```
Input:
- A set $\mathcal{B}^-$ containing all negative bags in the training set
- A bag $G$ containing all NIPs
- A candidate bag $B_i$ that contains $m_i$ instances $\mathbf{x}_{i,j}$ for $j = 1, \ldots, m_i$
- Trained bag-level SVM model $\Phi$
- An empty bag $\tilde{B}$
Output:
- Labels $y_{i,j} \in \{1, -1\}$ for instances $\mathbf{x}_{i,j} \in B_i$, for $j = 1, \ldots, m_i$

1:  Randomly sample negative instances from $\mathcal{B}^-$, and add them into $\tilde{B}$
2:  For $\mathbf{x}_{i,j} \in B_i$ do
3:      Compute Hausdorff distance $d(G, \mathbf{x}_{i,j})$ using Equation 5
4:  Sort $d(G, \mathbf{x}_{i,j})$ for $j = 1, \ldots, m_i$
5:  Find $\mathbf{x}_{i,j}$ with the minimum $d(G, \mathbf{x}_{i,j})$ in $B_i$
6:  Add $\mathbf{x}_{i,j}$ into $\tilde{B}$. Remove $\mathbf{x}_{i,j}$ from $B_i$
7:  Classify $\tilde{B}$ using $\Phi$
8:  If $\tilde{B}$ is negative
9:      $y_{i,j} = -1$
10: Else
11:     $y_{i,j} = 1$. Remove $\mathbf{x}_{i,j}$ from $\tilde{B}$
12: Goto step 5
```

3.3 Group Record Linkage

The MIL step may generate a number of false positive bags. In the context of group record linkage, this means that a group in one dataset is possibly matched to several groups in another dataset. For applications such as linking households in (historical) census data, a one-to-one linkage of groups is often required, e.g., to track the majority of members of a household across time. We therefore use the group linkage method proposed by On et al. [18] to reduce the number of multiple matches between groups. This method computes a similarity score between two groups, which is based on the number of record pairs that have been matched between two groups and the total number of records in the two groups. This is equivalent to selecting a bag that has been classified as positive in the MIL step, and using the instance labels to compute a similarity score for this bag. In [18], the similarity is calculated using the following normalised weight of the matched individual record pairs in the two groups:

$$\mathbb{S}_{i,j} = \frac{\sum_{(r_a, r_b) \in M} sim(r_a, r_b)}{m_i + m_j - |M|}, \tag{6}$$

where M is the set of record pairs matched between groups H_i and H_j, r_a and r_b are the records in the two groups, and m_i and m_j are the number of records in the two groups. The set of all links between H_i and H_j is the bag, and the link between r_a and r_b is one instance in this bag. Therefore, the similarity function $sim(r_a, r_b)$ can take on the label predicted by the MIL model, i.e. $sim(r_i^a, r_j^b) = 1$ for matched record pairs and $sim(r_i^a, r_j^b) = -1$ for non-matched pairs. This approach reduces the group linking problem to computing the Jaccard index between two groups [22]. A final set of matched groups is then extracted by selecting the group links with the highest similarity value $\mathbb{S}_{i,j}$ among all pairs of groups. When several group links generate the same highest value, all of them are considered as matches. Thus, the final output may still contain multiple links per group, but a much smaller number of them.

4 Experiments and Evaluation

We performed experiments on one synthetic dataset and six real census datasets using both the MILES and MILIS methods for the multiple instance learning step. For the implementation of MILES, we have used the MOSEK[1] system to solve the linear programming formulation in the one-norm SVMs. To train the MILIS algorithm, we have used LIBLINEAR [11]. The SVM regularisation parameter C was set using grid search on the training data. For Equation 3, we set $K = 10$ which is the same as in [14]. The feature mapping parameter γ in Equation 1 and the scale parameter β for the likelihood estimation in Equation 3 are both set to 1. For bag reconstruction in instance classification for the census data experiments, we have grouped a seed with 5 random negative instances. This is based on the fact that by average, a bag in the census datasets contains 5.65 instances, as can be calculated from Table 2.

For comparison purpose, we have implemented an alternative solution for bag and instance classification based on the group linkage method proposed by On et al. [18]. This method computes the sum of the similarity scores for each record pair, and then separates pairs into matches and non-matches by comparing the similarity sum with a threshold parameter ρ. The decision on the optimal ρ can be made based on the trade-off between the number of household pairs with multiple matches or unique matches. The matched households are then generated by grouping all matched record pairs that belong to the same matched household.

4.1 Synthetic Data Results

We have conducted experiments on synthetic data to evaluate the effectiveness of our instance classification method. The synthetic data generation follows the method in [14]. We randomly generated 1,000 positive instances and 5,000 negative instances, with each class generated from two Gaussian distributions. Then we constructed 50 positive bags by random sampling from both positive and negative instances, and 50 negative bags sampling from negative instances only. The number of instances in each bag is also randomly selected between 1 and 10. In this way, both bag and instance labels are known.

We split these bags into a training and a testing set, each containing 25 positive and 25 negative bags. We then trained bag-level classifiers using both the MILES and MILIS methods, and used them to classify instances in the testing set. This test is repeated 500 times over random partitions. The results show that the bag reconstruction method for instance classification presented in Section 3.2 is very effective. The random bag reconstruction method has achieved

[1] http://www.mosek.com

Table 1. Number of records and households in the historical census datasets

	1851	1861	1871	1881	1891	1901
Number of records	17,033	22,429	26,229	29,051	30,087	31,059
Number of households	3,295	4,570	5,575	6,025	6,379	6,848

Table 2. Number of bags and instances extracted from the historical census datasets

	1851–1861	1861–1871	1871–1881	1881–1891	1891–1901
Number of instances	2,104,171	2,200,876	2,459,272	3,043,786	3,318,738
Number of bags	325,921	441,355	472,239	494,270	588,436

an accuracy of 92.03 ± 2.21% using the MILES model, and 92.32 ± 2.43% using the MILIS model, while the greedy extension has achieved 92.89 ± 2.89% and 95.50 ± 2.47% on MILES and MILIS, respectively.

4.2 Historical Census Data Results

We used six census datasets from the district of Rawtenstall in the United Kingdom that were collected in ten-year intervals from 1851 to 1901. These census data contain twelve attributes per record, including the address, first and family name, age, gender, relationship to head, industry (occupation), and place of birth of each individual[2]. Because these data are of low quality, we have cleaned and standardised them using the *Febrl* data cleaning and record linkage system [7]. Details of this step can be found in Fu et al. [12]. Table 1 shows the number of records and households in each dataset.

The record level linkage was also conducted using *Febrl*. Instead of comparing all possible record pairs between two datasets, we used a traditional blocking technique combined with a Double-Metaphone encoding technique to index (block) the datasets [8]. We used a variety of approximate string comparison functions to calculate the similarity between individual record pairs following the approach given by Fu et al. [13]. The similarity scores calculated for a record pair were concatenated into a vector and then used in the MIL classification step.

We have manually labelled 1,000 household links from the 1871 and 1881 datasets, consisting of 500 matched and 500 non-matched households. To show the performance of the MILES and MILIS methods on household link classification, we performed 100-fold cross validation on the randomly split labelled data, with half used for training and half for testing.

Both the MILES and MILIS methods show similar performance, achieving 84.54 ± 1.33% and 83.75 ± 1.34% accuracy on household link classification, respectively. When efficiency is concerned, MILIS shows superior performance than MILES. The MILES method took 29.22 ± 6.37 seconds for training, and

[2] www.uk1851census.com

Table 3. Number of positive bags and instances classified in the different pairs of historical census datasets using the different methods described in this paper

	1851–1861	1861–1871	1871–1881	1881–1891	1891–1901
MILES-bag	7,728	9,644	9,705	9,650	12,583
MILIS-bag	8,832	11,369	9,870	9,175	11,282
Group-linkage-bag	47,249	50,494	49,306	48,212	50,058
MILES-random-instance	22,439	22,478	23,329	27,577	29,065
MILIS-random-instance	20,431	20,236	20,680	23,914	24,410
MILES-greedy-instance	22,063	21,771	23,170	27,019	28,987
MILIS-greedy-instance	20,738	21,436	22,228	25,050	24,872
Group-linkage-instance	67,122	67,340	65,528	65,595	67,483
After result fusion	775	1099	1484	1620	1689

0.88 ± 0.03 seconds for testing, while MILIS only took 2.17 ± 0.10 and 0.25 ± 0.04 seconds for each task. We did not evaluate the instance classification performance because the true record pair labels were not available to us.

In the next experiment, we re-trained the MILES and MILIS models using all the labelled data, and then classified all household and record links from any pair of consecutive census datasets, e.g. 1851 with 1861, 1861 with 1871, and so on. Because we were mainly interested in finding record matches in matched households, the instance classification was only performed on positively classified bags. As shown in Table 3, MILES and MILIS showed mixed performance on the bag-level classification, each having generated more positive bags than the counterpart on some datasets. By comparing the number of matched households with the total number of households in each census dataset (see Table 1), one can observe that the results contain multiple matches. This is expected because of two reasons. First, a household may split into several households, for example, due to the move-out of grown-up children, or two households might merge when widowed individuals form a new household. Second, there are many similar record pairs among different households, which may have generated false positive results. On the instance-level classification, the MILES-based models have consistently generated more positive instances than the MILIS-based models. The random bag reconstruction method, on the other hand, has achieved performance close to that of the greedy bag reconstruction method.

From Table 3, it can be observed that the group linkage method developed by On et al. [18] has generated many more household and record matches, i.e. more positive bags and instances, than the proposed MIL based methods. Statistics show that the MILES and MILIS based methods can reduce the number of matched bags in average between 79.98% and 79.40% respectively, when compared against the group linkage method described by On et al. [18]. Please note due to the lack of ground truth on household and record pair links, we have not used traditional measures such as accuracy and F-score for evaluation purposes.

We next applied the group linkage method introduced in Section 3.3 to reduce the number of multiple household matches, i.e. where one household is matched with multiple households. Figure 2 shows the performance of the proposed meth-

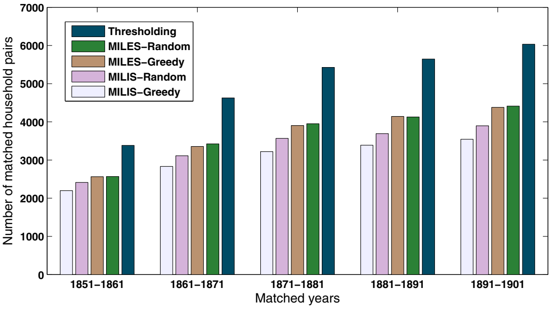


Fig. 2. Household matching results after group linkage step

ods and the thresholding method in [18]. The results indicate that the thresholding method generates the highest number of matches, followed by the MILES-based methods. The MILIS and greedy bag reconstruction combination has generated the smallest number of matches for all dataset pairs, which makes it the most reliable option in finding household matches between census datasets.

Finally, we performed results fusion so as to let the proposed methods vote for the most consistent household matches. This was performed by selecting household matches where all four options, i.e. MILES-random, MILES-greedy, MILIS-random, and MILIS-greedy, have agreed upon in their decision. These are the most reliable household matches that can be presented to researchers for further analysis. The last line in Table 3 shows the number of household matches after this fusion process.

5 Conclusion

We have introduced a group record linkage method based on multiple instance learning (MIL), and evaluated this method on real historical census data. In this method, group links are considered as bags and associated record links are treated as instances, with only the bag-level labels provided. The multiple instance learning paradigm has provided the group linkage problem with a suitable supervised learning tool to classify groups, even if the labels of record links are not available. We have shown the effectiveness of the proposed method on both synthetic and real historical census data from the UK.

In the future, we plan to extend the instance classification work so that instances selected for bag reconstruction better characterise the data distribution, and we will investigate approaches that allow linking records and households across several census datasets in an iterative fashion. We will also apply our method to other applications with a similar setting, such as bibliographic databases.

References

1. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: NIPS, Vancouver, Canada, pp. 561–568 (2003)

2. Bilenko, M., Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. In: ACM KDD, Washington, DC, pp. 39–48 (2003)
3. Bloothooft, G.: Multi-source family reconstruction. History and Computing 7(2), 90–103 (1995)
4. Chen, Y., Bi, J., Wang, J.: MILES: Multiple-instance learning via embedded instance selection. IEEE TPAMI 28(12), 1931–1947 (2006)
5. Chen, Y., Wang, J.Z.: Image categorization by learning and reasoning with regions. Journal of Machine Learning Research 5 (2004)
6. Christen, P.: Automatic Training Example Selection for Scalable Unsupervised Record Linkage. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 511–518. Springer, Heidelberg (2008)
7. Christen, P.: Development and user experiences of an open source data cleaning, deduplication and record linkage system. ACM SIGKDD Explorations 11(1), 39–48 (2009)
8. Christen, P.: A survey of indexing techniques for scalable record linkage and deduplication. IEEE TKDE (2011)
9. Dietterich, T.G., Lathrop, R.H., Lozano-Perez, T.: Solving the multiple-instance problem with axis-parallel rectangles. Artificial Intelligence 89, 31–71 (1997)
10. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate record detection: A survey. IEEE Transactions on Knowledge and Data Engineering 19(1), 1–16 (2007)
11. Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J.: LIBLINEAR: A library for large linear classification. JMLR 9, 1871–1874 (2008)
12. Fu, Z., Christen, P., Boot, M.: Automatic cleaning and linking of historical census data using household information. In: IEEE ICDM Workshop on DDDM (2011)
13. Fu, Z., Christen, P., Boot, M.: A supervised learning and group linking method for historical census household linkage. In: AusDM (2011)
14. Fu, Z., Robles-Kelly, A., Zhou, J.: MILIS: Multiple instance learning with instance selection. IEEE TPAMI 33(5), 958–977 (2011)
15. Fure, E.: Interactive record linkage: The cumulative construction of life courses. Demographic Research 3, 11 (2000)
16. Li, F., Sminchisescu, C.: Convex multiple instance learning by estimating likelihood ratio. In: NIPS (2010)
17. On, B.-W., Elmaciogl, E., Lee, D., Kang, J., Pei, J.: Improving grouped-entity resolution using quasi-cliques. In: IEEE ICDM, Hong Kong, pp. 1008–1015 (2006)
18. On, B.-W., Koudas, N., Lee, D., Srivastava, D.: Group linkage. In: IEEE ICDE, Istanbul, Turkey, pp. 496–505 (2007)
19. Quass, D., Starkey, P.: Record linkage for genealogical databases. In: ACM KDD Workshop, Washington, DC, pp. 40–42 (2003)
20. Reid, A., Davies, R., Garrett, E.: Nineteenth century Scottish demography from linked censuses and civil registers: a 'sets of related individuals' approach. History and Computing 14(1+2), 61–86 (2006)
21. Ruggles, S.: Linking historical censuses: a new approach. History and Computing 14(1+2), 213–224 (2006)
22. Tan, P., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson Addison-Wesley (2005)
23. Vapnik, V.: The Nature of Statistical Learning Theory. Springer (1995)
24. Xiao, Y., Liu, B., Cao, L., Yin, J., Wu, X.: SMILE: A similarity-based approach for multiple instance learning. In: IEEE ICDM, Sydney, pp. 309–313 (2010)

Incremental Set Recommendation Based on Class Differences

Yasuyuki Shirai[1], Koji Tsuruma[1], Yuko Sakurai[2], Satoshi Oyama[3], and Shin-ichi Minato[1,3]

[1] JST-ERATO MINATO Discrete Structure Manipulation System Project,
Hokkaido University, Sapporo, Japan
{shirai,tsuruma}@erato.ist.hokudai.ac.jp

[2] Graduate School of Information Science and Electrical Engineering,
Kyushu University, Fukuoka, Japan
ysakurai@inf.kyushu-u.ac.jp

[3] Graduate School of Information Science and Technology,
Hokkaido University, Sapporo, Japan
{oyama,minato}@ist.hokudai.ac.jp

Abstract. In this paper, we present a set recommendation framework that proposes sets of items, whereas conventional recommendation methods recommend each item independently. Our new approach to the set recommendation framework can propose sets of items on the basis on the user's initially chosen set. In this approach, items are added to or deleted from the initial set so that the modified set matches the target classification. Since the data sets created by the latest applications can be quite large, we use ZDD (Zero-suppressed Binary Decision Diagram) to make the searching more efficient. This framework is applicable to a wide range of applications such as advertising on the Internet and healthy life advice based on personal lifelog data.

Keywords: recommendation, classification, collaborative filtering, zero-suppressed binary decision diagram.

1 Introduction

Several techniques on information filtering and information recommendation such as collaborative filtering and content-based filtering have been reported [5][1][11]. In conventional collaborative filtering, items are recommended on the basis of their relevance to the user's preferences. Each item is recommended independently of the others; that is, the relationship of a recommended item to the other items is not considered.

In the real world, however, a user is often interested in a combination of items, such as the keywords in an advertisement and the places to be visited during a sightseeing tour. Recently proposed *set recommendation* techniques[12,10] consider the unit of recommendation to be a set of items and the constraints and requirements among them.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 183–194, 2012.

In this paper, we extend this approach to incorporate the use of an algorithm to present recommendations for modifying the user's initially chosen set. In our *incremental set recommendation* framework, it is assumed that each record (”item set”) in a database has been classified as a class such as positive/negative, and modifications are recommended that would change the item set so that it matched the target classification.

An example application of our framework is a recommendation system that uses a database in which the action history data for a group of people are stored. The data could be exercise history or dietary behavior, for example. Each person in the database is classified as either a success or failure w.r.t. to some target (e.g., weight loss). Those in the ”failure” group could use the system to obtain recommendations for specific behavior improvements that are based on the data for those in the ”success” group. The recommendations are made on the basis of the differences between the two groups and should change the user's actions and lifestyle as little as possible. ”Behavior improvements” for the exercise history example means the addition and/or deletion of item sets representing the type and amount of exercises performed, while for the dietary behavior example, it means the addition and/or deletion of item sets representing the type and quantity of food eaten. Another example application is a system for describing the items for sale on an Internet shopping site. The descriptions of poorly selling items would be modified on the basis of the descriptions of items that sell well.

The rest of the paper is organized as follows: Section 2 gives the basic definitions. In Section 3, we describe the implementation of our framework using a Zero-suppressed Binary Decision Diagram (ZDD) data structure. We present and discuss the results of its evaluation in Section 4. We conclude in Section 5 with a brief summary, some additional comments, and a mention of future work.

2 Definition

We will provide some definitions and notations as follows :

Definition 1 (Item). *An item is an atomic entity that represents a characteristic or feature and is denoted by a lower-case character, $a, b, c, \ldots$. A set of all items to be considered is denoted by Σ.*

In the exercise history example, each item could be the name of an exercise.

Definition 2 (Data Record and Class). *A data record is a collection of items that represent the attributes or characteristics of the target object (we use D to represent a data record). A class is a name for a set of data records, and is denoted by α, β, γ, ω or ϕ. Each data record belongs to only one class.*

“Positive” or “Negative” is an example of a class.

Definition 3 (Pattern Set/Class Membership). *A pattern set is a set of pairs, each of which consists of an item set and its weight (natural number). If the weight values are all the same, they can be omitted. A pattern set is denoted by C_ω where ω is the class identifier ($C_\omega = \{p : w_p | p \in 2^\Sigma, w_p \in \mathbb{N}\}$). If $q : w_q \in C_\omega$ (simply we write $q \in C_\omega$), q is called a pattern of class ω.*

Example : Let $C_\alpha = \{\{a,b,c\}:2, \{a,b,d\}:1, \{b,c,d\}:3\}$. $\{a,b,c\}$ is a pattern of class α, whereas $\{a,b\}, \{a,b,c,d\}$ and $\{b,d,e\}$ are not. We sometimes use polynomial notation for pattern set such as $C_\alpha = 2abc + abd + 3bcd$. A difference of two pattern set, such as $C_\alpha - C_\beta$, is defined as a difference of polynomials.

Definition 4 (Removable/Universal/Addable Items). *For a data record D, the items in D can be divided into removable items (D^-) and universal items (D^*). The recommendation algorithm can suggest the deletion of items only if they are elements of D^-. A set of addable items, denoted by D^+, is the set of items that can be added to D.*

Universal items intuitively correspond to essential features of the record. If no universal item is specified, $D^* = \emptyset$.

Definition 5 (Delete/Add-Constraints). *The upper bounds on the numbers of items that can be added or deleted for a data record D are denoted by N^D_{add}, N^D_{delete}, respectively.*

Definition 6 (Recommendation Candidate). *For a data record $D \in C_\phi (D \notin C_\omega)$, if there could be a candidate $D' \in C_\omega (D' \notin C_\phi)$ that is a modification of D by adding and/or deleting items under the conditions of N^D_{add} and N^D_{delete}, and if the weight of D' in $C_\omega - C_\phi$ is equal to or greater than the given natural number M (called weight condition), D' is called a recommendation candidate for class ω that satisfies N^D_{add}, N^D_{delete} and M.*

Example: Let $N^D_{add} = 1$ and $N^D_{delete} = 1$ for $D = \{a,b,f\}$ and weight condition $M = 2$ in the above example, $D' = \{a,b,c\}$ is a recommendation candidate for class C_α.

3 Set Recommendation Based on Class Differences

In general, the number of instances of each class could be quite large. For example, the number of articles for sale or the number of customers on a major Internet shopping site could reach several million. As another example, lifelog services using mobile devices generates enormous amount of data in recent years. To handle such huge numbers of data records, we use a ZDD (Zero-Suppressed Binary Decision Diagram) data structure. In this section, we first present an example of our set recommendation framework based on class differences. We then briefly introduce ZDD and our recommendation algorithm which uses the ZDD structure.

3.1 Example

Suppose we have pattern sets for classes α and β as follows:

$$C_\alpha = \{\{a,b,c\}:1, \{b,c\}:2, \{c\}:1, \{d,e\}:2, \{e\}:3\} \tag{1}$$

$$C_\beta = \{\{a,c\}:1, \{a,b,d\}:1\} \tag{2}$$

Suppose further that $\Sigma = \{a, b, c, d, e\}$ and $D = \{a, c\}$ ($D \in C_\beta$). The recommendation for D would consist of the following candidates (D') under the condition that $N^D_{add} = N^D_{delete} = 1$ and the weight condition $M = 1$: "Add b", "Delete a and add b", "Delete a". After those modification, D' would be identified as class α, rather than as class β. If we restrict the recommendation so that $M = 2$, we get only "Delete a and add b".

In this work, we use a VSOP (Valued-Sum Of Products) calculator based on ZDD for calculating the recommended items. We review ZDD and VSOP briefly in the next subsection.

3.2 ZDD and VSOP

Binary decision diagrams[2,4] (BDDs) are well-known and widely used for efficiently manipulating large-scale Boolean function data. A BDD is a directed graph representation of the Boolean function. The reduction rules in BDD consist of "node deletion rule" (delete all redundant nodes with two edges that point to the same node) and "node sharing rule" (share all equivalent sub-graphs).

ZDDs (Zero-suppressed BDDs) [6,4] are special type of BDDs which are suitable for implicitly handling large-scale combinatorial item set data. The reduction rules of ZDDs are slightly different from those of BDDs. They are illustrated in Fig. 1 (a).

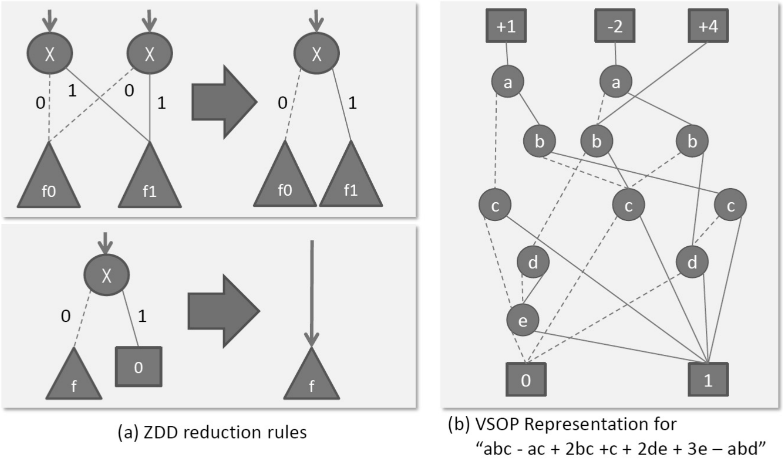
- Share equivalent nodes as well as ordinary BDDs.
- Delete all nodes whose 1-edge directly points to the 0-terminal node, and jump through to the 0-edge's destination.

ZDDs are especially more effective then BDDs for representing "sparse" combinations such as purchase history data. For instance, sets of combinations selecting 10 out of 1000 items can be represented by ZDDs up to 100 times more compactly than by ordinary BDDs.

VSOP (Valued-Sum-Of-Products Calculator)[7] is a program developed for calculating a combinatorial item set where each product term has a value, specified by symbolic expressions based on ZDD techniques. The value of each product can also be considered as a coefficient or a weight for each term. For example, the formula ($5abc + 3ab + 2bc + c$) represents a VSOP with four terms abc, ab, bc and c, each of which is valued as 5, 3, 2, and 1, respectively. VSOP supports numerical arithmetic operations based on Valued-Sum-Of-Products algebra, such as addition, subtraction, multiplication, division, and numerical comparison. The details of the algebra and arithmetic operations of a VSOP calculator are described in [6,7].

When dealing with integer values in binary coding, we have to consider the expression of negative numbers. VSOP adopted another binary coding[8] based on (-2), namely, each bit represents $1(= (-2)^0), -2(= (-2)^1), 4(= (-2)^2), -8(= (-2)^3), 16(= (-2)^4), \ldots$. For example, -12 can be decomposed into $(-2)^5 + (-2)^4 + (-2)^2$. In this encoding, each integer number as a coefficient can be uniquely represented.

Fig. 1 (b) shows the example of the VSOP representation for $abc - ac + 2bc + c + 2de + 3e - abd$. Since ac satisfies the top nodes labeled $+1$ and -2, the coefficient of item ac can be calculated by $+1 - 2 = -1$.



(a) ZDD reduction rules

(b) VSOP Representation for "abc - ac + 2bc +c + 2de + 3e – abd"

Fig. 1. ZDD Representation

3.3 Set Recommendation with ZDD Structure

In this subsection, we show the method for calculating a set of items to be recommended using ZDD. We first consider the following polynomials (valued sum of products) for (1) and (2) in Section 3.1:

$$C_\alpha - C_\beta = abc - ac + 2bc + c + 2de + 3e - abd$$

For a given D, we have to output a set of terms from $C_\alpha - C_\beta$, that are modification of D under the constraints of N_{add} and N_{delete}, and whose coefficients are equal to or larger than given integer M.

Fig. 2 shows an example search process on the ZDD structure for $C_\alpha - C_\beta$, where $N_{add} = N_{delete} = 1$ and $D = ac$. In this figure, the search process starts with each top node $+1, -2, +4$ respectively and then item sets satisfying the constraints are extracted for each top node $+1, -2, +4$. The pair of numbers for item addition and deletion is attached to each edge, as shown in Fig. 2. If the pair does not satisfy condition N_{add} or N_{delete}, searching along that path is terminated. For example, since the pair on the edge from c (left side in Fig. 2) is $(0, 2)$, which does not satisfy the N_{delete} condition, searching along the path below that node is terminated.

Item sets that need to be found under the condition of $M = 2$ must satisfy one of $(0, 0, 1), (1, 0, 1), (0, 1, 1)$, or $(1, 1, 1)$ for the top nodes $(+1, -2, +4)$ in Fig. 2. For example, suppose $D = ac, N_{add} = 1$ and $N_{delete} = 1$. Since the

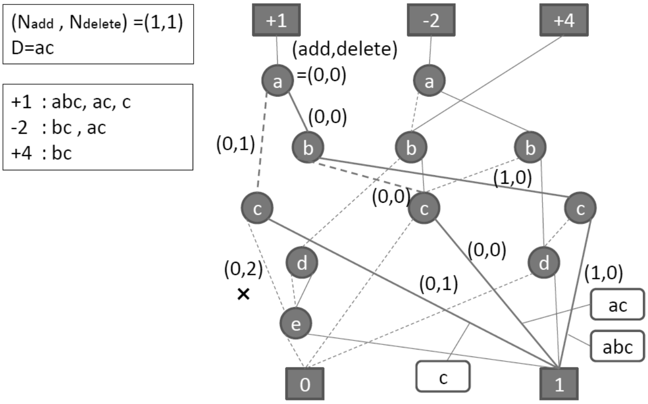


Fig. 2. Search on ZDD Structure

numbers of added items and deleted items w.r.t *bc* are 1 and 1 respectively, and since *bc* satisfies $(0, 1, 1)$ for the top nodes $(+1, -2, +4)$, *bc* is a recommendation candidate under the condition of $M = 2$. As another example, since the numbers of added items and deleted items w.r.t *abc* are 1 and 0 respectively, and since the candidates *abc* satisfies $(1, 0, 0)$ for the top nodes $(+1, -2, +4)$, *abc* could be a recommendation candidate under the condition of $M = 1$ as well as *bc* described above. By the same way, *c* is also a recommendation candidate under the condition of $M = 1$

The naive search algorithm on a ZDD structure is shown in Algorithm 1.

4 Experiments

We first evaluate the efficiency of our approach based on ZDD, using artificial data, and then we show the examples using actual Internet application data.

4.1 Performance Evaluation

The problem we provided for performance evaluation in this experiment consists of 170 items in total ($|\Sigma| = 170$), and each record contains 5 items. There are two classes: positive and negative.

There are two execution scenarios: one used a random data set, and the other used a fixed pattern data set. In the random data set, items occurs randomly in each record; in the fixed pattern data set, fixed positive and negative patterns were prepared (20 patterns for each class), and each pattern consisted of three items.

In the fixed pattern data set, three items in each record are taken from the fixed patterns, and two are taken from the random patterns. In actual applications, such as an Internet purchase history, there would be some fixed patterns

Algorithm 1. Naive Search Algorithm on ZDD structure

```
Given $D$ (target items), $N_{add}, N_{delete}$ (upper limits of number of "add" items, "delete" items), $M(\geq 1)$ (weight condition), $ZDD$ (ZDD structure whose top nodes are $1, \ldots, N$ (ex. -1,-2,4,...)).

for $i = 1$ to $N$ do
   initialize $(path_i = \{\}, AddList = \{\}, DeleteList = \{\})$
   $L = L + get\_candidate(path_i, i, AddList, DeleteList)$
end for
$merge\_output(L, M)$ (merge the results for each node $(1, \ldots, n)$ and output the results whose coefficients $\geq M$)

Function $get\_candidate(path, n, AddList, DeleteList)$ $\{n$ is a node of $ZDD\}$
if $|DeleteList| > N_{delete}$ or $|AddList| > N_{add}$ then
   return $null$
else if $n$ is a terminal node 1 then
   return $path$
else if $n$ is a terminal node 0 then
   return $null$
else
   let $n_0$, $n_1$ : node which is connected by 0 edge, 1 edge of node $n$, respectively.
   if $n \in D$ then
      $get\_candidate(path, n_0, AddList, DeleteList + \{n\})$
      $get\_candidate(path + \{n\}, n_1, AddList, DeleteList)$
   else
      $get\_candidate(path, n_0, AddList, DeleteList)$
      $get\_candidate(path + \{n\}, n_1, AddList + \{n\}, DeleteList)$
   end if
end if
End Function
```

in both classes. In this experiment, we used three data set sizes for each class: 1, 5, and 10 million records.

The system was implemented in Java, and the experiments were run on SUSE Linux Enterprise Server 10 with a quad-core AMD Opteron 3 GHz CPU, and 512 GB RAM. The execution times are shown in Table 1. The times shown are an average for ten trials. In the tables, "sequential search" in which all items in each record were ordered and stored in memory was done for comparison.

With the random data sets, there were no substantial differences between the ZDD-based search and the sequential search. This is because a ZDD data structure is not much more compact or efficient rather than a flat data structure. The number of ZDD nodes for the random data sets were respectively 2696527, 11789288, and 20738481. Their relationship is almost linear. In contrast, with the fixed pattern data sets, there were marked differences between the two searches. The ZDD-based search was more efficient due to the compact representation by ZDD. The numbers of ZDD nodes for the fixed pattern data sets were respectively

Table 1. Experimental Results for Performance Evaluation

(a) For Random Data Set (Time : msec)

	ZDD-based Search			Sequential Search		
	1M	5M	10M	1M	5M	10M
$N_{add} = 1, N_{delete} = 1, M = 1$	13	12	17	63	255	514
$N_{add} = 2, N_{delete} = 2, M = 1$	16	27	47	70	349	652
$N_{add} = 3, N_{delete} = 3, M = 1$	72	326	522	92	496	1152
$N_{add} = 3, N_{delete} = 3, M = 2$	74	308	521	93	513	1123
$N_{add} = 3, N_{delete} = 3, M = 3$	73	328	541	98	502	1163

(b) For Fixed Pattern Data Set (Time : msec)

	ZDD-based Search			Sequential Search		
	1M	5M	10M	1M	5M	10M
$N_{add} = 1, N_{delete} = 1, M = 1$	6	5	7	72	279	564
$N_{add} = 2, N_{delete} = 2, M = 1$	6	8	9	79	384	609
$N_{add} = 3, N_{delete} = 3, M = 1$	7	9	13	94	450	784
$N_{add} = 3, N_{delete} = 3, M = 2$	7	9	17	95	419	769
$N_{add} = 3, N_{delete} = 3, M = 3$	7	9	15	94	424	776

313594, 363112, and 377979. These data sets were relatively small and did not linearly increase in size. This was reflected in the total execution times.

In actual applications, there are usually fixed patterns in item occurrences for each class. Although actual application data are not as strongly biased as in our experiment, we can nevertheless conclude that the ZDD-based search approach is well suited for actual applications.

4.2 Example : Internet Shopping Advertising

As one Internet application, we used data from Rakuten Ichiba [13], the largest online shopping mall in Japan, with over 30,000 online stores (September 2009). The company has released some of their data for use in academic research[13].

We investigated the relationship between article descriptions at the time of article introduction for sale and the number of user reviews attached to each article. The descriptions had been written (in Japanese) by a person working for the shop where the article was to be sold. Article descriptions are important because they attract customers through on-line searching.

The data fields in the original data are shop code, purchase article id, article name, article description, price, category, and number of reviews. There are 31 top categories in total. We used data labeled with the category "Ladies Fashion" and "Japanese Sake" (liquor).

In order to define characteristics from each description, we first extracted the nouns, adjectives, verbs, and adverbs from the descriptions using the Japanese language morphological analysis program called Chasen[1]. After some modifi-

[1] http://chasen.naist.jp

cations (n-gram word concatenation to generate collocation, word selection by $TF \times IDF$ measurement, etc.), the item sets (Σ) were defined. The explanatory variables for each article consists of the occurrences of the selected words in the article descriptions. The class of the article was determined by the number of reviews. That is, we classified an article “positive” if it had more than two reviews and “negative” if it had no reviews.

In the Japanese Sake category, there were 517 items in total ($|\Sigma| = 517$), 3085 positive records, and 3372 negative records. Each record generally had five to ten items. We set N_{delete} and N_{add} to respectively 2 and 4. In the Ladies Fashion category, there were 1475 items in total ($|\Sigma| = 1475$), 3166 positive records, and 7194 negative records. Both N_{delete} and N_{add} were set to 3.

Table 2 shows some of the results translated from the original Japanese. We assumed that all items in D (original item set) were removable (i.e., it did not contain any universal items) and set weight condition M to 1. For the Japanese Sake category, we found that keywords that created attractive images were preferred rather than technical keywords such as “rice malt” and “carefully screened.” These results reflect the Internet shopping situation for Japanese Sake; that is, people who like to buy Japanese sake on the Internet generally put more emphasis on the image or feeling of drinking sake rather than the technical details, unlike those who buy it in actual shops. For the Ladies Fashion category, keywords giving specific descriptions of each article, such as “tiered skirt,” “hemline,” and “shoulder strap adjustment” were preferred to ones that created a visual image of their usage or that described the technical details. That is, people shopping on the Internet for fashion prefer specific images and specifications, unlike people shopping in actual shops.

4.3 Example : AOL Search Logs

As the other Internet application, we used data from AOL’s Search Log Collection [14]. This collection consists of about 21 million web search queries input by about 650,000 AOL users from March to May 2006. The records include ‘Query,’ ‘ItemRank,’ and ‘ClickURL’ (the last two items were included only if the user had clicked on a search result).

We used records in which there was a ‘ClickURL’ entry and the ‘ItemRank’ was less than 5 as the positive data and those without a ‘ClickURL’ entry as negative data. The objective of this analysis was to present sets of items to be deleted from or added to the original queries so as to increase the likelihood that the user would click on a search result.

The results are shown in Table 3 (words preceded by “*” are universal items). We found that, if the user wants to find specific information about travel, it would be better to add a specific place-name such as europe, south africa, or italy. From the last example in the table, the keyword ‘cheap’ would not be adequate if the user wanted a reasonable price. Better keywords would be ‘discounts,’ ‘best,’ and ’packages.’

Table 2. Example : Internet Shopping Advertising

Category	Original	Added Items	Deleted Items
Sake	rice malt, production area, box, gift	plum brandy, woman	rice malt, production area
	low temperature, slow, distillation, actual producer, distilled spirit, flavor , production area	rich, black malt, tasty	low temperature, slow, distillation
	river-bed water, carefully-screened, actual producer, distilled spirit, characteristics, production area, flavor, tasty	deepness, barley distilled spirit	river-bed water, carefully-screened, characteristics
	cold storage, representative, refined sake, barm, acid degree	bright, limited, flavor	representative, refined sake
	recommend, cold storage, barm, father, acid degree	wonderful, brand sake	recommend, father
Ladies	skirt, casual, polyurethane, hip, real scale	tiered skirt, appeal	casual, polyurethane, hip
	shopping, travel, event, import	hemline, casual	shopping
	love, casual, travel, event, import	shoulder strap adjustment, beautiful leg	love, casual, travel
	boot-cut, straight, pants, polyurethane, silhouette, body	camel-hair, autumn and winter	pants

These results shows that our recommendation flamework can suggest possible candidates for query modifications, in order to get more appropriate search results for users.

5 Summary and Future Works

In this paper, we have described a new approach to the set recommendation problem: changes (item addition and deletion) to a set of items are recommended on the basis of class differences. Since recommendation services are becoming more and more popular, our framework should be effective for actual applications rather than simply being used for collaborative filtering. The use of our algorithms, which use the ZDD data structure, results in efficient calculation for huge data sets, especially when the data is biased, as it generally is in actual applications. Although we only considered the case of two classes for simplicity, we can easily consider a case in which there are three or more classes or there are multiple classification criteria for the input data.

In related work, Dong et al.[3] proposed using an *emerging patterns* approach to detecting differences in classes and using a classification framework based on the emerging patterns. While frequent pattern mining generally cannot detect the characteristic item pattern for each class, their approaches focus on detecting

Table 3. Example : AOL Search Logs

Original	Added Items	Deleted Items
adventure,*travel,blogs	tours,student	blogs
	africa	blogs
	family	blogs
	italy	adventure
	south,africa	adventure
*family,travel,vacations	europe	vacations
	packages,rome	vacations
	best	travel
	cheap	travel
	packages	travel
cheap,*europe,*vacation	discounts	cheap
	best	cheap
	packages	cheap

item sets that are meaningful for classification. Although their motivation is very similar to ours, they have not yet reported a recommendation procedure based on emerging patterns.

Other researchers have developed set recommendation procedures based on certain constraints such as recommendation costs, orders and other conditions [12,10]. These procedures are practically applicable to trip advice and university course selection, for example. Although we do not assume any constraint between items as pre-defined knowledge, incorporating such constraints into our recommendation framework should improve its effectiveness.

Searching under the constraints described in this paper is closely related to searching based on the Levenshtein distance (edit distance). Efficient algorithms, such as dynamic programming approach, have been developed to calculate the distance, and many implementations including approximation approaches have been introduced [9]. The problem we focused on in this paper is slightly different from those for the edit distance. Our problem is to find similar items from given polynomials under the constraints of a limited number of item additions and deletions with a weight constraint. A comparison of the problems remains for future work.

Future work also includes extending our results in several directions :

- The search procedure based on the ZDD structure described in this paper still contains redundant processes. Efficient search strategies such as using a cache of pre-searched results need to be investigated.
- We assume in this framework that items occur only positively in patterns. In actual applications, however, “don’t care” plays an important role in recommendation. We need to investigate ways to incorporate such items.

Acknowledgment. We are grateful to Rakuten, Inc., for providing the Internet shopping data used in this research.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Bryant, R.E.: Graph-based algorithms for Boolean function manipulation. IEEE Transactions on Computers 35(8) (August 1986)
3. Dong, G., Zhang, X., Wong, L., Li, J.: CAEP: Classification by Aggregating Emerging Patterns. In: Arikawa, S., Nakata, I. (eds.) DS 1999. LNCS (LNAI), vol. 1721, pp. 30–42. Springer, Heidelberg (1999)
4. Knuth, D.E.: The Art of Computer Programming. Bitwise Tricks & Techniques, vol. 4(1), pp. 117–126. Addison-Wesley (2009)
5. Melville, P., Sindhwani, V.: Recommender Systems. In: Encyclopedia of Machine Learning, pp. 829–838. Springer (2010)
6. Minato, S.: Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems. In: Proc. of 30th ACM/IEEE Design Automation Conference, DAC 1993 (1993)
7. Minato, S.: VSOP (Valued-Sum-of-Products) Calculator for Knowledge Processing Based on Zero-Suppressed BDDs. In: Jantke, K.P., Lunzer, A., Spyratos, N., Tanaka, Y. (eds.) Federation over the Web. LNCS (LNAI), vol. 3847, pp. 40–58. Springer, Heidelberg (2006)
8. Minato, S.: Implicit Manipulation of Polynomials Using Zero-Suppressed BDDs. In: Proc. of IEEE The European Design and Test Conference (1995)
9. Navarro, G.: A Guided Tour to Approximate String Matching. ACM Computing Surveys 33(1) (2001)
10. Parameswaran, A., Venetis, P., Garcia-Molina, H.: Recommendation Systems with Complex Constraints: A Course Recommendation Perspective. Transactions on Information Systems 29(4) (2011)
11. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. In: Advances in Artificial Intelligence (2009)
12. Xie, M., Lakshmanan, L.V.S., Wood, P.T.: Breaking out of the box of recommendations: From Items to Packages. In: Proc. of the 4th ACM Conf. on Recommender Systems (2010)
13. (Rakuten data disclosure), http://rit.rakuten.co.jp/rdr/index_en.html
14. (AOL search logs), http://www.gregsadetsky.com/aol-data

Active Learning for Cross Language Text Categorization

Yue Liu, Lin Dai*, Weitao Zhou, and Heyan Huang

School of Computer Science and Technology
Beijing Institute of Technology, Beijing 100081, China
{kerwin,dailiu,zhouwt,hhy63}@bit.edu.cn

Abstract. Cross Language Text Categorization (CLTC) is the task of assigning class labels to documents written in a target language (e.g. Chinese) while the system is trained using labeled examples in a source language (e.g. English). With the technique of CLTC, we can build classifiers for multiple languages employing the existing training data in only one language, therefore avoid the cost of preparing training data for each individual language. One challenge for CLTC is the culture differences between languages, which causes the classifier trained on the source language doesn't perform well on the target language. In this paper, we propose an active learning algorithm for CLTC, which takes full advantage of both labeled data in the source language and unlabeled data in the target language. The classifier first learns the classification knowledge from the source language, and then learns the cultural dependent knowledge from the target language. In addition, we extend our algorithm to double viewed form by considering the source and target language as two views of the classification problem. Experiments show that our algorithm can effectively improve the cross language classification performance.

Keywords: Cross Language Text Categorization, Active Learning.

1 Introduction

Due to the explosive growth of electronic documents in different languages, there's an urgent need for effective multilingual text organizing techniques. Cross Language Text Categorization (CLTC) is the task of assigning class labels to documents written in a target language (e.g. Chinese) while the system is trained using labeled examples in a source language (e.g. English). With the technique of CLTC, we can build classifiers for multiple languages employing the existing training data in only one language, thereby avoiding the cost of preparing training data for each individual language.

The basic idea under CLTC is the documents in different languages may share the same semantic information [14], although they're in different representations. Previous works on CLTC have tried several methods to erase the language barrier and show promising results. However, despite language barrier, there's another problem for CLTC. That is the differences between cultures, which may cause

* Corresponding author.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 195–206, 2012.

topic drift between languages. For example, news of the category *sports* from China (in Chinese) and US (in English) may concern different topics. The former may talk more about table tennis and Liu Xiang while the later may prefer NBA and NFL. As a result, even if the language barrier is perfectly erased, some knowledge of the target language still can't be learned from the training data in the source language. This will inevitably affect the performance of categorization. To solve this problem, making use of the unlabeled data in the target language will be helpful. Because these data is often easy to obtain and contains knowledge of the target language. If we can provide techniques to learn from it, the resulting classifier is expected to get more fit for the target language, thereby give better categorization performance.

In this paper, we propose an active learning algorithm for cross language text categorization. Our algorithm makes use of both labeled data in the source language and unlabeled data in the target language. The classifier first learns the classification knowledge from the source language, and then learns the cultural dependent knowledge from the target language. In addition, we extend our algorithm to double viewed form by considering the source and target language as two views of the classification problem. Experiments show that our algorithm can effectively improve the cross language classification performance. To the best of our knowledge, this is the first study of applying active learning to CLTC.

The rest of the paper is organized as follows. First, related works are reviewed in Section 2. Then, our active learning approach for CLTC is presented in Section 3 and its extension to double viewed form is introduced in Section 4. Section 5 presents the experimental results and analysis. Finally, Section 6 gives conclusions and future work.

2 Related Work

Several previous works have addressed the task of CLTC. [2] proposes practical approaches based on machine translation. In their work, two translation strategies are considered. The first strategy translates the training documents into the target language and the second strategy translates the unlabeled documents into the source language. After translation, monolingual text categorization is performed. [12] introduces a model translation method, which transfers classification knowledge across languages by translating the model features and takes into account the ambiguity associated with each word. Besides translation, in some other studies multilingual models are learned and used for CLTC, such as the multilingual domain kernels learned from comparable corpora [5] and the multilingual topic models mined from Wikipedia [8]. Moreover, there are also some studies of using lexical databases (e.g. WordNet) for CLTC [1].

All the previous methods have somehow solved the language barrier between training documents and unlabeled documents. But only a few have considered the culture differences between languages. In these works, authors try to solve this problem by employing some semi-supervised learning techniques. [12] employs a self-training process after the model translation. This process applies the translated model to predict a set of unlabeled documents in target language and iteratively chooses the most confident classified documents to retrain the model. As a result, the model is adapted to better fit the target language. [9] proposes an EM based training algorithm. It consists of an initialization step that trains a classifier using translated labeled documents and an iteration step that repeats the E and M phases. In the E phase, the classifier predicts the unknown labels of a collection of documents in the target language. In the M phase, these documents with labels obtained in E phase are used to estimate the parameters of the new classifier. [16] investigates the use of co-training in cross language sentiment categorization. In their work, Chinese and English features are considered as two independent views of the categorization problem.

The common idea of above methods is to automatically label and use the documents in target language. To reduce the noises introduced by classification errors, the documents with low prediction certainty are usually underutilized. In this paper, we consider such documents to contain important information and will explore them through our active learning algorithm.

3 Active Learning for CLTC

3.1 Cross Language Text Categorization

Given a collection TR_e of labeled documents in language E and a collection TS_c of unlabeled documents in language C, in the scenario of CLTC, we would like to train a classifier using TR_e to organize the documents in TS_c. E and C is usually referred as the source language and target language respectively. In this paper, we suppose E is English and C is Chinese. In practice, they can be replaced with any other language pairs.

To solve the language barrier between training and test documents, we can employ a machine translation tool. The translation can be performed in two directions: the first direction translates all training documents into Chinese and the second direction translates all test documents into English. Both approaches convert the cross language problem into monolingual one. In this section, we choose the training set translation approach. First, we translate TR_e into Chinese, denoting it by $TR_{e\text{-}c}$. Then, we learn a classifier $C_{e\text{-}c}$ based on $TR_{e\text{-}c}$. Suppose the translation process gives accurate enough results, $C_{e\text{-}c}$ obtains the classification knowledge transferred from English.

$C_{e\text{-}c}$ can be applied to the unlabeled Chinese documents directly. Since the documents of same topic in different languages may share some common semantics, $C_{e\text{-}c}$ may be able to make very certain predications for some Chinese documents by the classification knowledge transferred from English. However, for some other documents, $C_{e\text{-}c}$ may get confused, as the class-discriminative information of these documents can't be detected. The latter case is usually caused by culture differences. For instance, a classifier trained using English *sports* samples may not be able to recognize Liu Xiang, a famous Chinese hurdle athlete, in a Chinese document. We can then make an observation that Chinese documents, which are uncertain to be classified by $C_{e\text{-}c}$, usually contain culture dependent classification knowledge that can't be learnt from the translated training data. From this observation, we derive the active learning algorithm to improve $C_{e\text{-}c}$.

3.2 Apply Active Learning to CLTC

Active learning [11] is a form of learning algorithm for situations in which unlabeled data is abundant but labeling data is expensive. In such a scenario, the learner actively selects examples from a pool of unlabeled data and asks the teacher to label.

In the context of CLTC, we can assume an additional collection U_c of unlabeled documents in target language (Chinese in this paper) is available, since the unlabeled data is usually easy to obtain. Our algorithm consists of two steps. In the first step, we train a classifier using the translated training set $TR_{e\text{-}c}$, this classifier can be considered as an initial learner which has learnt the classification knowledge transferred from the source language. In the second step, we apply this classifier to the documents in U_c and select out the documents with lowest classification certainty. Such documents are expected to contain most culture dependent classification knowledge. We label them and put them into the training set. Consequently, the classifier is re-trained. The second step is repeated for several iterations, in order to let the classifier learn the culture dependent knowledge from the target language. Figure 1 illustrates the whole process.

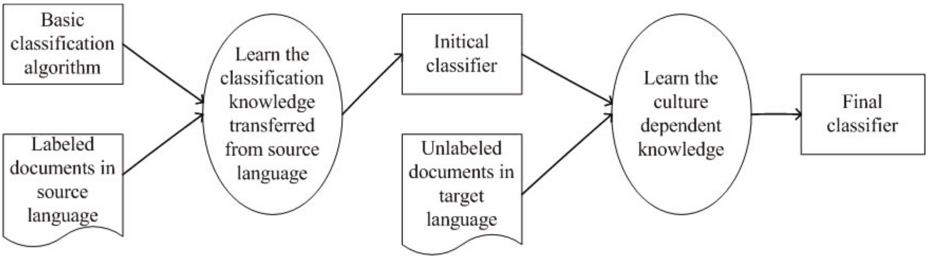

Basic classification algorithm
Labeled documents in source language
Learn the classification knowledge transferred from source language
Initical classifier
Unlabeled documents in target language
Learn the culture dependent knowledge
Final classifier

Fig. 1. The active learning process

In our approach, a basic classification algorithm is required to train the initial classifier. We employ support vector machine, as it has been well studied in previous work for active learning [15,6,11]. Note that our algorithm is independent on specific classification techniques.

Given the translated labeled set $TR_{e\text{-}c}$, each example can be represented as (x, y), where $x \in R^p$ is the feature vector and $y \in \{1, 2, \ldots k\}$ is the corresponding class label. A classifier learnt from $TR_{e\text{-}c}$ can predict the unknown class label for a document d in U_c. To measure the prediction certainty, we can refer to the membership probabilities of all possible classes.

However, SVM can't give probabilistic outputs directly. Some tricks have been proposed in [7]. For binary-class SVM, given the feature vector $x \in R^p$, and the label $y \in \{-1, 1\}$, the membership probability $p(y = 1|x)$ can be approximated using a sigmoid function,

$$P(y = 1|x) = 1/(1 + exp(Af(x) + B)), \tag{1}$$

where $f(x)$ is the decision function of SVM, A and B are parameters to be estimated. Maximum likelihood estimation is used to solve for the parameters,

$$\begin{gathered} \min_{(A,B)} - \sum_{i=1}^{l} (t_i \log p_i + (1 - t_i) \log (1 - p_i)), \\ \text{where,} \\ p_i = \frac{1}{1 + exp(Af(x_i) + B)}, \\ t_i = \begin{cases} \frac{N_+ + 1}{N_+ + 2} & \text{if } y_i = 1; \\ \frac{1}{N_- + 2} & \text{if } y_i = -1. \end{cases} \end{gathered} \tag{2}$$

N_+ and N_- are the number of positive and negative examples in the training set. Newton's method with backtracking line search can be used to solve this optimization problem [7]. For multi-class SVM, we can obtain the probabilities through pair coupling [18]. Suppose that r_{ij} is the binary probability estimate of $P(y = i|y = i \; or \; j, x)$, and p_i is the probability $P(y = i|x)$, the problem can be formulated as

$$\begin{gathered} min_p \frac{1}{2} \sum_{i=1}^{k} \sum_{j, j \neq i} (r_{ji} p_i - r_{ij} p_j)^2, \\ subject \; to \; \sum_{i=1}^{k} p_i = 1 \text{ and } p_i \geq 0, \forall i, \end{gathered} \tag{3}$$

where k denotes the number of classes. This optimization problem can be solved using a direct method such as Gaussian elimination, or a simple iterative algorithm [18].

In practice, we employ the toolbox $LibSVM$ [4], which is widely used in data mining tasks [13]. It implements the above methods for multi-class probability estimation. After obtaining the class membership probabilities of a document, we use the best against second best (BVSB) approach [6] to estimate the classification certainty. This approach has been demonstrated to be effective for multi class active learning task [6]. It measures the certainty by the difference between the probability values of the two classes having the highest estimated probabilities. The larger the difference, the higher the certainty is. Suppose c is the classifier, d is the document to be classified, i and j are the two classes with highest probabilities, then we calculate the certainty score using

$$Certainty(d, c) = P(y = i|d, c) - P(y = j|d, c). \tag{4}$$

Based on the discussions above, we describe the proposed algorithm in Algorithm 1.

Algorithm 1. Active learning algorithm for CLTC

```
Input:
    The labeled set in the source language, $TR_e$;
    The unlabeled set in the target language, $U_c$;
Output:
    Classifier $C$;
Translate examples in $TR_e$ into the target language, and denote the translated set
    by $TR_{\text{e-c}}$
Let $TrainingSet = TR_{\text{e-c}}$
Repeat $I$ times:
    Train classifier $C$ using $TrainingSet$
    Classify documents in $U_c$ by $C$ and measure the prediction certainty using
        Equation 4
    Let $S$ be a set of n documents with the lowest prediction certainty
    Remove $S$ from $U_c$
    Label documents in $S$ by the teacher
    Add the newly labeled examples to $TrainingSet$
Return $C$
```

4 Double Viewed Active Learning

In this section, we extend our algorithm to double viewed form. In chief, the source and target language are considered as two views of the classification problem. The same idea was utilized in [16].

4.1 Two Views of the Problem

In Section 3, we convert the cross language problem into monolingual one with the help of a machine translation tool. As illustrated in Figure 2, the translation can be performed in two directions. The first direction translates the training set TR_e into Chinese and the second direction translates both the test set TS_c and additional unlabeled set U_c into English. Each direction gives us a monolingual view of the problem. In Section 3, we apply our active learning algorithm based on the Chinese view. In this section, we will show how to take advantage of both views and extend our algorithm to double viewed active learning.

First, we perform the translation following both directions. As a result, each document is associated with two views: the Chinese view and the English view. We denote the double viewed training set, test set and additional unlabeled set by TR, TS and U respectively. Then, two initial classifiers are trained using TR based on Chinese view and English view individually. We apply both of them to the unlabeled set U.

Since predictions made by the two classifiers are based on individual views of one document, they may have different certainties. As illustrated in Figure 3, the pool of unlabeled documents is split into four regions. In region C, the English classifier is certain on the documents, while the Chinese classifier is not. In region D, it's the opposite. For these scenarios, we can employ a co-training

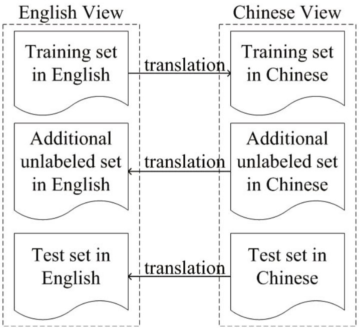


Fig. 2. Two directions of translation

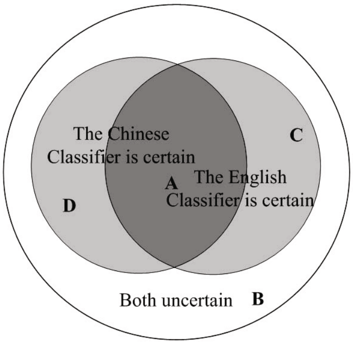

The Chinese Classifier is certain
C
A
The English Classifier is certain
D
Both uncertain
B

Fig. 3. Certainty distribution over the unlabeled documents

[3] approach, which labels documents according to the confident classifier and generate new training examples for the unconfident one. In other words, the two learners can teach each other in some times, needn't always ask the teacher. Based on this idea, we present the double viewed active learning algorithm in the next section.

4.2 Double Viewed Active Learning

Given a document d and two classifiers C_e and C_c, we measure whether both classifiers are certain about its prediction by the average certainty,

$$Average\_Certainty(d, C_e, C_t) = (Certainty(d, C_e) + Certainty(d, C_c))/2. \quad (5)$$

To measure whether a classifier is more certain than the other, we refer to the difference between their certainties,

$$Certainty\_Difference(d, C_e, C_c) = Certainty(d, C_e) - Certainty(d, C_c). \quad (6)$$

Our double viewed active learning algorithm is described by Algorithm 2.

After the learning phase, we get two classifiers C_e and C_c. As a result, in the classification phase we can obtain two predictions for a document. Since both classifiers output class membership probabilities, they can be combined in the following way to give the overall prediction,

$$P(y = i|x) = (P(y = i|x, C_c) + P(y = i|x, C_e))/2. \quad (7)$$

5 Evaluation

5.1 Experimental Setup

We choose English-Chinese as our experimental language pair. English is regarded as the source language while Chinese is regarded as the target language.

Algorithm 2. Double viewed active learning algorithm for CLTC

```
Input:
    The labeled set in the source language, $TR_e$;
    The unlabeled set in the target language, $U_c$;
Output:
    Classifiers $C_e$ and $C_c$;
Generate two-view labeled set $TR$ by translate $TR_e$ into the target language
Generate two-view unlabeled set $U$ by translate $U_c$ into the source language
Let $TrainingSet = TR$
Repeat $I$ times:
    Train classifier $C_e$ using $TrainingSet$ based on the source language view
    Train classifier $C_c$ using $TrainingSet$ based on the target language view
    Classify $U$ by $C_e$ and $C_c$ respectively
    Let $S$ be the $n$ documents from $U$ having lowest $Average\_Certainty(d)$
    Let $L$ be the documents from $U$ having $Certainty(d, C_c) > h$ or $Certainty(d, C_e) > h$, where $h$ is the certainty threshold
    Let $E_e$ and $E_c$ be $m$ documents from $L$ having highest $Certainty\_Difference(d, C_e, C_c)$ and $Certainty\_Difference(d, C_c, C_e)$ respectively
    Remove $E_e$, $E_c$ and $S$ from $U$
    Label $E_e$ and $E_c$ according to $C_e$ and $C_c$ respectively; Label $S$ by the teacher
    Add $E_e$, $E_c$, $S$ to $TrainingSet$
Return $C_e$ and $C_c$
```

Since there is not a standard evaluation benchmark available for cross language text categorization, we build a data set from the Internet. This data set contains 42610 Chinese and English news pages during the year 2008 and 2009, which fall into eight categories: Sports, Military, Tourism, Economy, Information Technology, Health, Autos and Education. The main content of each page is extracted and saved in plain text.

In our experiments, we select 1000 English documents and 2000 Chinese documents from each class. The set of English documents is treated as the training set TR_e. For the Chinese documents, we first randomly select 1000 documents from each class to form the test set TS_c, and leave the remaining documents as the additional unlabeled set U_c.

As we will use the two views of each document in our algorithm, we employ *Google Translate* [1] to translate all Chinese documents into English and all English documents into Chinese. Then, for all Chinese or Chinese translated documents, we segment the text with the tool *ICTCLAS* [2], afterwards remove the common words. For all English or English translated documents, the *EuropeanLanguageLemmatizer* [3] is applied to restore each word in the text to its base form. Then we use a stop words list to eliminate common words.

Each document is transformed into an English feature vector and a Chinese feature vector with *TF-IDF* format. The *LibSVM* package is employed for the

[1] http://translate.google.com

[2] http://ictclas.org/

[3] http://lemmatizer.org/

basic classifier. We choose linear kernel due to its good performance in text classification task. Since we need probabilistic outputs, the b option of $LibSVM$ is selected for both training and classification. The cost parameter c is set to 1.0 as default. We use Micro-Average F1 score as the evaluation measure, as it's a standard evaluation used in most previous categorization research [10,17].

5.2 Results and Discussions

In this section, we present and discuss the experimental results of the proposed algorithms.

Single Viewed Active Learning. In the first experiment, we would like to verify the effectiveness of our active learning algorithm described in Algorithm 1. An initial classifier is trained using the translated labeled set $TR_{\text{e-c}}$ and then applied to the Chinese unlabeled set U_c. In each iteration, 10 documents with the lowest prediction certainty are selected and labeled by the teacher. To validate this selecting strategy, we also implement another strategy which selects 10 documents randomly for comparison. In each iteration, a new classifier is retrained on the expanded labeled set and its performance is evaluated on the testing set TS_c. The corresponding micro average F1 curves are plotted in Figure 4.

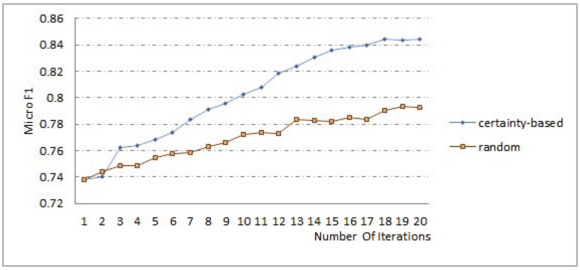


Fig. 4. Micro-F1 curves of single viewed algorithm

We can observe that, the initial classifier doesn't perform well on the Chinese test set. As the number of iterations increases, the performance is significantly improved. The certainty-based strategy shows an obvious advantage over the random strategy. This verify our assumption that documents with low prediction certainty usually contain culture dependent classification knowledge and therefore are most informative for the learner. After 20 iterations, the Micro average F1 measure on the 8000 test documents is increased by about 11 percents while the additional cost is to label 200 selected examples.

Double Viewed Active Learning. In the following experiments, we verify the double viewed algorithm described in Algorithm 2. First, two initial classifiers are trained using the labeled set based on English and Chinese view individually. Then the active learning process is performed. We set the parameter n to 10, which means in each iteration 10 examples having lowest average certainty are selected and labeled by the teacher; and we set m to 5, which means each classifier labels 5 examples for the other. The certainty threshold h is set to 0.8, in order to reduce the error introduced by automatically labeled examples. In each iteration, the two classifiers are retrained and applied to the test set. We combine their predictions based on each view to get the overall prediction. Figure 5 shows the micro average F1 curves of the Chinese, English and overall classifiers. The curve of the single viewed algorithm is plotted as well for comparison.

We can observe that, the English classifier generally has better performance than the Chinese one, a possible reason is that more noises are introduced in Chinese view due to the text segmentation process. The overall classifier has highest accuracy, as it combines the information from both views. All the three classifiers generated by double viewed algorithm outperform the one of the single viewed algorithm. Because in each iteration they get 10 more labeled examples (each classifier automatically labels 5 examples for the other).

In our double viewed algorithm, the classifiers learn from each other and the teacher. We would like to investigate the effect of the two approaches individually. This can be done by set the parameter n and m in Algorithm 2. We first set n to 10 and m to 0, then set n to 0 and m to 5. The corresponding curves are showed in Figure 6.

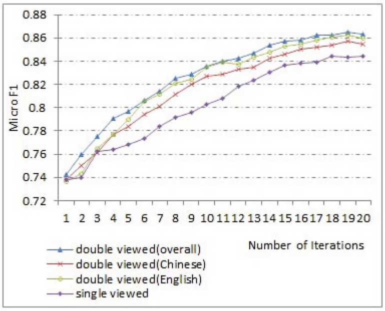


Fig. 5. Micro-F1 curves of double viewed algorithm

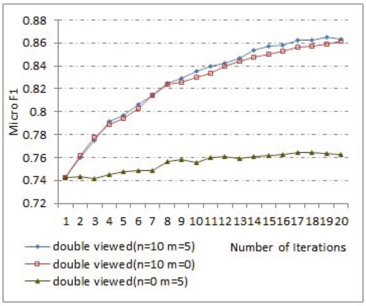


Fig. 6. Compare effects of the two learning approaches

As we can see, learning from the teacher makes a significant contribution to the improvement of the performance, while the effect of learning from the partner is weaker. The latter maybe caused by two reasons: first, there may be some errors introduced by the automatically labeled examples; second, since the Chinese and English views of one document are not completely independent, the C and D region illustrated in Figure 2 may be very limited. However, learning from the partner is still helpful, and it reduces the labor of the teacher to achieve the same performance.

Table 1. Comparison of different methods

Category	ML			MTE			MTC			Single Viewed (n=10, 20 iterations)			Double Viewed (n=10,m=5,20 iterations, overall)		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
sports	94.0	89.3	91.6	86.2	72.1	78.5	83.6	73.8	78.4	86.5	85.5	86	84.3	91.2	87.6
military	92.9	94.2	93.5	68.7	85.2	76.1	67.1	84.7	74.9	78.5	93.7	85.4	82.0	95.8	88.4
tourism	87.4	91.8	89.5	72.3	68.2	70.2	75.7	71.2	73.4	83.2	85.4	84.3	85.8	83.9	84.8
economy	87.5	87.2	87.3	88.4	56.2	68.7	85.2	62.7	72.2	84.2	80.4	82.3	83.9	87.8	85.8
IT	90.4	90.3	90.3	72.1	80.2	75.9	71.6	75.2	73.4	86.9	85.7	86.3	91.7	83.3	87.3
health	92.1	91.2	91.6	69.1	81.4	74.7	73.2	79.7	76.3	85.0	87.3	86.1	87.6	85.6	86.6
autos	93.7	91.4	92.5	65.3	88.5	75.2	62.5	89.8	73.7	85.1	92.6	88.7	89.7	92.3	91.0
education	88.4	90.1	89.2	87.9	58.1	70.0	88.9	53.9	67.1	87.8	64.7	74.5	86.3	70.5	77.6

Comparison. In Table 1, we present the detailed classification results of our algorithms, comparing with two basic machine translation based methods. The first one, denoted as **MTC**, translates the training set TR_e into Chinese and trains a classifier; the second one, denoted as **MTE**, trains a classifier in English and translates the test set TS_c into English. In addition, we also build a monolingual classifier (**ML**) by using all documents in U_c as training data. The **ML** method plays the role of an upper-bound, since the best classification results are expected when monolingual training data is available.

We can observe that, the **ML** classifier has the best performance as expected, since it's trained on the labeled data in the target language, so that there's no drawback caused by language barrier or cultural differences. Comparing with the two basic machine translation methods **MTE** and **MTC**, our active learning algorithms, both single viewed and double viewed, significantly improve the classification performance of each class. The double viewed algorithm has better performance than the single viewed one, as it combines the information from both views and makes use of the automatically labeled examples.

6 Conclusions and Future Works

In this paper, we proposed the active learning algorithm for cross language text categorization. The proposed method can effectively improve the cross language classification performance by learning from unlabeled data in the target language. For the future work, we will incorporate more metrics in the selecting strategy of active learning. For instance, can we detect the scenario in which the classifier is pretty certain but actually wrong? If such examples can be detected and labeled for retraining, the classifier will be further adaptable for the target language.

Acknowledgments. This work is supported by National Natural Science Foundation of China (60803050, 61132009)and BIT Team of Innovation.

References

1. Amine, B.M., Mimoun, M.: Wordnet based cross-language text categorization. In: 2007 IEEE/ACS International Conference on Computer Systems and Applications, pp. 848–855. IEEE (2007)
2. Bel, N., Koster, C.H.A., Villegas, M.: Cross-Lingual Text Categorization. In: Koch, T., Sølvberg, I.T. (eds.) ECDL 2003. LNCS, vol. 2769, pp. 126–139. Springer, Heidelberg (2003)
3. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pp. 92–100. ACM (1998)
4. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
5. Gliozzo, A., Strapparava, C.: Cross language text categorization by acquiring multilingual domain models from comparable corpora. In: Proceedings of the ACL Workshop on Building and Using Parallel Texts, pp. 9–16. Association for Computational Linguistics (2005)
6. Joshi, A.J., Porikli, F., Papanikolopoulos, N.: Multi-class active learning for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 2372–2379. IEEE (2008)
7. Lin, H.T., Lin, C.J., Weng, R.C.: A note on platt's probabilistic outputs for support vector machines. Machine Learning 68(3), 267–276 (2007)
8. Ni, X., Sun, J.T., Hu, J., Chen, Z.: Mining multilingual topics from wikipedia. In: Proceedings of the 18th International Conference on World Wide Web, pp. 1155–1156. ACM (2009)
9. Rigutini, L., Maggini, M., Liu, B.: An EM based training algorithm for cross-language text categorization. In: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 529–535. IEEE (2005)
10. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys (CSUR) 34(1), 1–47 (2002)
11. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2009)
12. Shi, L., Mihalcea, R., Tian, M.: Cross language text classification by model translation and semi-supervised learning. In: Proc. EMNLP, pp. 1057–1067. Association for Computational Linguistics, Cambridge (2010)
13. Tang, J., Liu, H.: Feature selection with linked data in social media. In: SIAM International Conference on Data Mining (2012)
14. Tang, J., Wang, X., Gao, H., Hu, X., Liu, H.: Enriching short texts representation in microblog for clustering. Frontiers of Computer Science (2012)
15. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. The Journal of Machine Learning Research 2, 45–66 (2002)
16. Wan, X.: Co-training for cross-lingual sentiment classification. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, vol. 1, pp. 235–243. Association for Computational Linguistics (2009)
17. Wang, X., Tang, J., Liu, H.: Document clustering via matrix representation. In: The 11th IEEE International Conference on Data Mining, ICDM 2011 (2011)
18. Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multi-class classification by pairwise coupling. The Journal of Machine Learning Research 5, 975–1005 (2004)

Evasion Attack of Multi-class Linear Classifiers

Han Xiao[1,2], Thomas Stibor[2], and Claudia Eckert[2]

[1] CeDoSIA of TUM Graduate School
[2] Chair for IT Security
Technische Universität München, Germany
{xiaoh,stibor,claudia.eckert}@in.tum.de

Abstract. Machine learning has yield significant advances in decision-making for complex systems, but are they robust against adversarial attacks? We generalize the evasion attack problem to the multi-class linear classifiers, and present an efficient algorithm for approximating the optimal disguised instance. Experiments on real-world data demonstrate the effectiveness of our method.

1 Introduction

Researchers and engineers of information security have successfully deployed systems using machine learning and data mining for detecting suspicious activities, filtering spam, recognizing threats, etc. [2,12]. These systems typically contain a classifier that flags certain instances as malicious based on a set of features. Unfortunately, evaded malicious instances that fail to be detected are inevitable for any known classifier. To make matters worse, there is evidence showing that adversaries have investigated several approaches to evade the classifier by disguising malicious instance as normal instances. For example, spammers can add unrelated words, sentences or even paragraphs to the junk mail for avoiding detection of the spam filter [11]. Furthermore, spammers can embed the text message in an image. By adding varied background and distorting the image, the generated junk message can be difficult for OCR systems to identify but easy for humans to interpret [7]. As a reaction to adversarial attempts, authors of [5] employed a cost-sensitive game theoretic approach to preemptively adapt the decision boundary of a classifier by computing the adversary's optimal strategy. Moreover, several improved spam filters that are more effective in adversarial environments have been proposed [7,3].

The ongoing war between adversaries and classifiers pressures machine learning researchers to reconsider the vulnerability of classifier in adversarial environments. The problem of evasion attack is posed and a query algorithm for evading linear classifiers is presented [10]. Given a malicious instance, the goal of the adversary is finding a disguised instance with the minimal cost to deceive the classifier. Recently, the evasion problem has been extended to the binary convex-inducing classifiers [13].

We continue investigate the vulnerability of classifiers to the evasion attack and generalize this problem to the family of multi-class linear classifiers; e.g. linear support vector machines [4,6,9]. Multi-class linear classifiers have become one of the most promising learning techniques for large sparse data with a huge number of instances and features. We propose an adversarial query algorithm for searching minimal-cost

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 207–218, 2012.

disguised instances. We believe that revealing a scar on the multi-class classifier is the only way to fix it in the future. The contributions of this paper are:

1. We generalize the problem of evasion attack to the multi-class linear classifier, where the instance space is divided into multiple convex sets.
2. We prove that effective evasion attack based on the linear probing is feasible under certain assumption of the adversarial cost. A description of the vulnerability of multi-class linear classifiers is presented.
3. We propose a query algorithm for disguising an adversarial instance as any other classes with minimal cost. The experiment on two real-world data set shows the effectiveness of our algorithm.

2 Problem Setup

Let $\mathcal{X} = \{(x_1, \ldots, x_D) \in \mathbb{R}^D \,|\, L \leq x_d \leq U \text{ for all } d\}$ be the *feature space*. Each component of an *instance* $\mathbf{x} \in \mathcal{X}$ is a *feature* bounded by L and U which we denote as x_d. A basis vector of the form $(0, \ldots, 0, 1, 0, \ldots, 0)$ with a 1 only at the d^{th} feature terms $\boldsymbol{\delta}_d$. We assume that the feature space representation is known to the adversary, thus the adversary can query any point in $\mathcal{X}$.

2.1 Multi-class Linear Classifier

The target classifier f is a mapping from feature space $\mathcal{X}$ to its response space $\mathcal{K}$; i.e. $f : \mathcal{X} \to \mathcal{K}$. We restrict our attention to *multi-class linear classifiers* and use $\mathcal{K} = \{1, \ldots, K\}, K \geq 2$ so that

$$f(\mathbf{x}) = \underset{k}{\operatorname{argmax}}\ \mathbf{w}_k \mathbf{x}^{\mathrm{T}} + b_k, \tag{1}$$

where $k = 1, \ldots, K$ and $\mathbf{w}_k \in \mathbb{R}^D, b_k \in \mathbb{R}$. Decision boundaries between class k and other classes are characterized by $\mathbf{w}_k$ and b_k. We assume that $\mathbf{w}_1, \ldots, \mathbf{w}_K$ are linearly independent. The classifier f partitions $\mathcal{X}$ into K sets; i.e. $\mathcal{X}_k = \{\mathbf{x} \in \mathcal{X} \,|\, f(\mathbf{x}) = k\}$.

2.2 Attack of Adversary

As a motivating example, consider a text classifier that categorizes incoming emails into different topics; e.g. sports, politics, lifestyle, spam, etc. An advertiser of pharmacological products is more likely to disguise the spam as lifestyle rather than politics in order to attract potential consumers while remaining inconspicuous.

We assume the adversary's attack will be against a fixed f so the learning method of decision boundaries and the training data used to establish the classifier are irrelevant. The adversary does not know any parameter of f but can observe $f(\mathbf{x})$ for any $\mathbf{x}$ by issuing a *membership query*. In fact, there are a variety of domain specific mechanisms that an adversary can employ to observe the classifier's response to a query. Moreover, the adversary is only aware of an adversarial instance $\mathbf{x}^{\mathrm{A}}$ in some class, and has no information about instances in other classes. This differs from previous work which require at least one instance in each binary class [10,13]. In practice, $\mathbf{x}^{\mathrm{A}}$ can be seen as the most desired instance of adversary; e.g. the original spam. The adversary attempts to disguise $\mathbf{x}^{\mathrm{A}}$ so that it can be recognized as other classes.

2.3 Adversarial Cost

We assume that the adversary has the access to an *adversarial cost function* $a(\mathbf{x}, \mathbf{y}) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{0+}$. An adversarial cost function measures the distance between two instances $\mathbf{x}, \mathbf{y}$ in $\mathcal{X}$ from the adversary's prospective. We focus on a linear cost function which measures the weighted ℓ_1 distance so that

$$a(\mathbf{x}, \mathbf{y}) = \sum_{d=1}^{D} e_d |x_d - y_d|, \tag{2}$$

where $0 < e_d < \infty$ represents the cost coefficient of the adversary associates with the d^{th} feature, allowing that some features may be more important than others. In particular, given the adversarial instance $\mathbf{x}^{\text{A}}$, function $a(\mathbf{x}, \mathbf{x}^{\text{A}})$ measures different costs of using some instances as compared to others. Moreover, we use $\mathcal{B}(\mathbf{y}, C) = \{\mathbf{x} \in \mathcal{X} \mid a(\mathbf{x}, \mathbf{y}) \leq C\}$ to denote the cost ball centered at $\mathbf{y}$ with cost no more than C.

In generalizing work [10], we alter the definition of *minimal adversarial cost* (MAC). Given a fixed classifier f and an adversarial cost function a we define the MAC of class k with respect to an instance $\mathbf{y}$ to be the value

$$\text{MAC}(k, \mathbf{y}) = \min_{\mathbf{x}:\mathbf{x} \in \mathcal{X}_k} a(\mathbf{x}, \mathbf{y}), \quad k \neq f(\mathbf{y}).$$

2.4 Disguised Instances

We now introduce some instances with special adversarial cost that the adversary is interested in. First of all, instances with cost of $\text{MAC}(k, \mathbf{y})$ are termed *instances of minimal adversarial cost* (IMAC), which is formally defined as

$$\text{IMAC}(k, \mathbf{y}) = \{\mathbf{x} \in \mathcal{X}_k \mid a(\mathbf{x}, \mathbf{y}) = \text{MAC}(k, \mathbf{y}), k \neq f(\mathbf{y})\}.$$

Ideally, the adversary attempts to find $\text{IMAC}(k, \mathbf{x}^{\text{A}})$ for all $k \neq f(\mathbf{x}^{\text{A}})$. The most naive way for an adversary to find the IMAC is performing a brute-force search. That is, the adversary randomly samples points in $\mathcal{X}$ and updates the best found instance repetitively. To formulate this idea, we further extend the definition of IMAC. Assume $\widetilde{\mathcal{X}}$ is the set of adversary's sampled or observed instances so far and $\widetilde{\mathcal{X}} \subset \mathcal{X}$, we define *instance of sample minimal adversarial cost* (ISMAC) of class k with respect to an instance $\mathbf{y}$ to be the value

$$\text{ISMAC}(k, \mathbf{y}) = \operatorname*{argmin}_{\mathbf{x}:\mathbf{x} \in \widetilde{\mathcal{X}} \cap \mathcal{X}_k} a(\mathbf{x}, \mathbf{y}), \quad k \neq f(\mathbf{y}).$$

Note, that in practice the exact decision boundary is unknown to the adversary, thus finding exact value of IMAC becomes an infeasible task. Nonetheless, it is still tractable to approximate IMAC by finding ϵ-IMAC, which is defined as follows

$$\epsilon\text{-IMAC}(k, \mathbf{y}) = \{\mathbf{x} \in \mathcal{X}_k \mid a(\mathbf{x}, \mathbf{y}) \leq (1 + \epsilon) \cdot \text{MAC}(k, \mathbf{y}), k \neq f(\mathbf{y}), \epsilon > 0\}.$$

That is, every instance in $\epsilon\text{-IMAC}(k, \mathbf{y})$ has the adversarial cost no more than a factor of $(1 + \epsilon)$ of the $\text{MAC}(k, \mathbf{y})$. The goal of the adversary now becomes finding $\epsilon\text{-IMAC}(k, \mathbf{x}^{\text{A}})$ for all classes $k \neq f(\mathbf{x}^{\text{A}})$ while keeping ϵ as small as possible.

3 Theory of Evasion Attack

We discuss the evasion attack from a theoretical point of view. Specifically, by describing the feature space as a set of convex polytopes, we show that IMAC must be attained on the convex surface. Under a reasonable assumption of adversarial cost function, effective evasion attack can be performed by linear probing. Finally, we derive bounds for quantitatively studying the vulnerability of multi-class linear classifiers to linear probing.

Lemma 1. *Let* $\mathcal{X}_k = \{\mathbf{x} \in \mathcal{X} \mid f(\mathbf{x}) = k\}$*, where the classifier* f *is defined in (1). Then* $\mathcal{X}_k$ *is a closed convex polytope.*

Proof. Let $\mathbf{x}$ be a point in $\mathcal{X}_k$. As $\mathbf{x} \in \mathcal{X}$ it follows that

$$\mathbf{x}^{\mathrm{T}} \geq L \cdot \mathbb{1}_D \qquad \text{and} \qquad -\mathbf{x}^{\mathrm{T}} \geq U \cdot \mathbb{1}_D, \tag{3}$$

where $\mathbb{1}_D$ is a D-dimensional unit vector $(1, \dots, 1)$. Moreover, since $f(\mathbf{x}) = k$, it follows that

$$\begin{pmatrix} \mathbf{w}_k - \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_k - \mathbf{w}_K \end{pmatrix} \mathbf{x}^{\mathrm{T}} \geq \begin{pmatrix} b_1 - b_k \\ \vdots \\ b_K - b_k \end{pmatrix}. \tag{4}$$

Thus, the foregoing linear inequalities define an intersection of at most $(K + 2D - 1)$ half-spaces. Denote $H_i^+ = \{\mathbf{x} \in \mathcal{X} \mid \widetilde{\mathbf{w}_i}\mathbf{x}^{\mathrm{T}} \geq \widetilde{b_i}\}$, where $1 \leq i \leq (K + 2D - 1)$. We have $\mathcal{X}_k = \bigcap_i H_i^+$, which establishes a half-space representation of convex polytope [8,14]. □

Lemma 1 indicates that the classifier f decomposes $\mathbb{R}^D$ into K convex polytopes. Following the notations and formulations introduced in [8], we represent a hyperplane H_i as the boundary of a half-space ∂H_i^+; i.e. $H_i = \partial H_i^+ = \{\mathbf{x} \in \mathcal{X} \mid \widetilde{\mathbf{w}_i}\mathbf{x}^{\mathrm{T}} = \widetilde{b_i}\}$. Let $\mathcal{X}_k = \bigcap_{p=1}^{P_k} H_p^+$, where $\{H_1^+, \dots, H_{P_k}^+\}$ is *irredundant*[1] to $\mathcal{X}_k$. Let $\mathcal{H}_k = \{H_1^+, \dots, H_{P_k}^+\}$ be an irredundant set that defines $\mathcal{X}_k$, then $\mathcal{X}_k \subset \operatorname{int} \mathcal{X}$ provided that none half-space in $\mathcal{H}_k$ is defined by (3). Moreover, we define the p^{th} *facet* of $\mathcal{X}_k$ as $F_{kp} = H_p \cap \mathcal{X}_k$, and the *convex surface* of $\mathcal{X}_k$ as $\partial\mathcal{X}_k = \bigcup_{p=1}^{P_k} F_{kp}$.

Theorem 1. *Let* $\mathbf{y}$ *be an instance in* $\mathcal{X}$ *and* $k \in \mathcal{K} \setminus f(\mathbf{y})$*. Let* $\mathbf{x}$ *be an instance in IMAC*$(k, \mathbf{y})$ *as defined in Section 2.3. Then* $\mathbf{x}$ *must be attained on the convex surface* $\partial\mathcal{X}_k$*.*

Proof. We first show the existence of IMAC$(k, \mathbf{y})$. By Lemma 1, $\mathcal{X}_k$ defines a feasible region. Thus minimizing $a(\mathbf{x}, \mathbf{y})$ on $\mathcal{X}_k$ is a solvable problem. Secondly, $\mathcal{X}_k$ is bounded in each direction of the gradient of $a(\mathbf{x}, \mathbf{y})$, which implies that IMAC$(k, \mathbf{y})$ exists.

We now prove that $\mathbf{x}$ must lie on $\partial\mathcal{X}_k$ by contrapositive. Assume that $\mathbf{x}$ is not on $\partial\mathcal{X}_k$ thus is an interior point; i.e. $\mathbf{x} \in \operatorname{int} \mathcal{X}_k$. Let $\mathcal{B}(\mathbf{y}, C)$ denote the ball centered at $\mathbf{y}$ with cost no more than $a(\mathbf{x}, \mathbf{y})$. Due to the convexity of $\mathcal{X}_k$ and $\mathcal{B}(\mathbf{y}, C)$, we have $\operatorname{int} X_k \cap \operatorname{int} \mathcal{B}(\mathbf{y}, C) \neq \emptyset$. Therefore, there exists at least one instance in $\mathcal{X}_k$ with cost less than $a(\mathbf{x}, \mathbf{y})$, which implies that $\mathbf{x}$ is not IMAC$(k, \mathbf{y})$. □

[1] Let $\mathcal{C}$ be a convex polytope such that $\mathcal{C} = \bigcap_{i=1}^{n} H_i^+$. The family $\{H_1^+, \dots, H_n^+\}$ is called *irredundant* to $\mathcal{C}$ provided that $\bigcap_{1 \leq j \leq n, j \neq i} H_j^+ \neq \mathcal{C}$ for each $j = 1, \dots, n$.

Theorem 1 restricts the searching of IMAC to the convex surface. In particular, when cost coefficients are equal, e.g. $e_1 = \cdots = e_D$, we can show that searching in all axis-aligned directions gives at least one IMAC.

Theorem 2. *Let* $\mathbf{y}$ *be an instance in* $\mathcal{X}$ *such that* $\mathcal{X}_{f(\mathbf{y})} \subset \operatorname{int} \mathcal{X}$. *Let* P *be the number of facets of* $\mathcal{X}_{f(\mathbf{y})}$ *and* F_p *be the* p^{th} *facet, where* $p = \{1, \ldots, P\}$. *Let* $G_d = \{\mathbf{y} + \theta \boldsymbol{\delta}_d \,|\, \theta \in \mathbb{R}\}$, *where* $d \in \{1, \ldots, D\}$. *Let* $\mathcal{Q} = \{G_d \cap F_p \,|\, d = 1, \ldots, D, p = 1, \ldots, P\}$, *in which each element differs from* $\mathbf{y}$ *on only one dimension. If the adversarial cost function defined in (2) has equal cost coefficients, then there exists at least one* $\mathbf{x} \in \mathcal{Q}$ *such that* $\mathbf{x}$ *is IMAC*$(f(\mathbf{x}), \mathbf{y})$.

Proof. Let H_p be the hyperplane defining the p^{th} facet F_p. Consider all the points of intersection of the lines G_d with the hyperplanes H_p; i.e. $\mathcal{I} = \{G_d \cap H_p \,|\, d = 1, \ldots, D, p = 1, \ldots, P\}$. Let $\mathbf{x} = \operatorname{argmin}_{\mathbf{x} \in \mathcal{I}} \; a(\mathbf{x}, \mathbf{y})$. Then $\mathbf{x}$ is our desired instance.

We prove that $\mathbf{x} \in \mathcal{Q}$ by contrapositive. Suppose $\mathbf{x} \notin \mathcal{Q}$, due to the convexity of $\mathcal{X}_{f(\mathbf{y})}$, the line segment $[\mathbf{x}, \mathbf{y}]$ intersects $\partial \mathcal{X}_{f(\mathbf{y})}$ at a point on another facet. Denote this point as $\mathbf{z}$, then $\mathbf{z}$ differs from $\mathbf{y}$ on only one dimension and $a(\mathbf{z}, \mathbf{y}) < a(\mathbf{x}, \mathbf{y})$.

Next, we prove $\mathbf{x}$ is IMAC$(f(\mathbf{x}), \mathbf{y})$ by contrapositive. Let $\mathcal{B}(\mathbf{y}, C)$ denote the *regular* cost ball centered at $\mathbf{y}$ with cost no more than $a(\mathbf{x}, \mathbf{y})$. That is, each vertex of the cost ball has the same distance of C with $\mathbf{y}$. Suppose $\mathbf{x}$ is not IMAC$(f(\mathbf{x}), \mathbf{y})$, then there exists $\mathbf{z} \in \mathcal{X}_{f(\mathbf{x})} \cap \operatorname{int} \mathcal{B}(\mathbf{y}, C)$. By Theorem 1, $\mathbf{z}$ and $\mathbf{x}$ must lie on the same facet, which is defined by a hyperplane H^*. Let $\mathcal{Q}^*$ be intersection points of H^* with lines $G_1, \ldots, G_D$; i.e. $\mathcal{Q}^* = \{G_d \cap H^* \,|\, d = 1, \ldots, D\}$. Then there exists at least one point $\mathbf{v} \in \mathcal{Q}^*$ such that $\mathbf{v} \in \operatorname{int} \mathcal{B}(\mathbf{y}, C)$. Due to the regularity of $\mathcal{B}(\mathbf{y}, C)$, we have $a(\mathbf{v}, \mathbf{y}) < a(\mathbf{x}, \mathbf{y})$. □

We now define special convex sets for approximating ϵ-IMAC near the convex surface. Given $\epsilon > 0$, the interior parallel body of $\mathcal{X}_k$ is $\mathcal{P}_{-\epsilon}(k) = \{\mathbf{x} \in \mathcal{X}_k \,|\, \mathcal{B}(\mathbf{x}, \epsilon) \subseteq \mathcal{X}_k\}$ and the corresponding exterior parallel body is defined as $\mathcal{P}_{+\epsilon}(k) = \bigcup_{\mathbf{x} \in \mathcal{X}_k} \mathcal{B}(\mathbf{x}, \epsilon)$. Moreover, the interior margin of $\mathcal{X}_k$ is $\mathcal{M}_{-\epsilon}(k) = \mathcal{X}_k \setminus \mathcal{P}_{-\epsilon}(k)$ and the corresponding exterior margin is $\mathcal{M}_{+\epsilon}(k) = \mathcal{P}_{+\epsilon}(k) \setminus \mathcal{X}_k$. By relaxing the searching scope from the convex surface to a margin in the distance ϵ, Theorem 1 and Theorem 2 immediately imply the following results.

Corollary 1. *Let* $\mathbf{y}$ *be an instance in* $\mathcal{X}$ *and* $k \in \mathcal{K} \setminus f(\mathbf{y})$. *For all* $\epsilon > 0$ *such that* $M_{-\epsilon}(k) \neq \emptyset$, ϵ*-IMAC*$(k, \mathbf{y}) \subseteq \mathcal{M}_{-\epsilon}(k)$.

Corollary 2. *Let* $\mathbf{y}$ *be an instance in* $\mathcal{X}$ *and* ϵ *be a positive number such that* $\mathcal{P}_{+\epsilon}(f(\mathbf{y})) \subset \operatorname{int} \mathcal{X}$. *Let* P *be the number of facets of* $\mathcal{P}_{+\epsilon}(f(\mathbf{y}))$ *and* F_p *be the* p^{th} *facet, where* $p = \{1, \ldots, P\}$. *Let* $G_d = \{\mathbf{y} + \theta \boldsymbol{\delta}_d \,|\, \theta \in \mathbb{R}\}$, *where* $d \in \{1, \ldots, D\}$. *Let* $\mathcal{Q} = \{G_d \cap F_p \,|\, d = 1, \ldots, D, p = 1, \ldots, P\}$, *in which each element differs from* $\mathbf{y}$ *on only one dimension. If adversarial cost function defined in (2) has equal cost coefficients, then there exists at least one* $\mathbf{x} \in \mathcal{Q}$ *such that* $\mathbf{x}$ *is in* ϵ*-IMAC*$(f(\mathbf{x}), \mathbf{y})$.

Corollary 1 and Corollary 2 point out an efficient way of approximating ϵ-IMAC with linear probing, which forms the backbone of our proposed algorithm in Section 4.

Finally, we consider the vulnerability of a multi-class linear classifier to linear probing. The problem arises of detecting convex polytopes in $\mathcal{X}$ with a random line. As one can easily scale any hypercube to a unit hypercube with edge length 1, our proof is restricted to the unit hypercube in $\mathbb{R}^D$.

Definition 1 (Vulnerability to Linear Probing). *Let $\mathcal{X} = [0,1]^D$, and $\mathcal{X}_1, \ldots, \mathcal{X}_K$ be the sets that tile $\mathcal{X}$ according to the classifier $f : \mathcal{X} \to \{1, \ldots, K\}$, with $K \geq 2$. Let G be a random line in $\mathbb{R}^D$ that intersects $\mathcal{X}$. Denote Z the number of sets intersect G, the vulnerability of classifier f to linear probing is measured by the expectation of Z.*

When $\mathbb{E}\, Z$ is small, a random line intersects small number of decision regions and not much information is leaked to the adversary. Thus, a robust multi-class classifier that resists linear probing should have a small value of $\mathbb{E}\, Z$.

Theorem 3. *Let f be the multi-class linear classifier defined in (1), then the expectation of Z is bounded by $1 < \mathbb{E}\, Z < 1 + \frac{\sqrt{2}(K-1)}{2D}$.*

Proof. By Lemma 1, we have K convex polytopes $\mathcal{X}_1, \ldots, \mathcal{X}_K$. Let $\mathcal{F}$ be the union of all facets of polytopes. Observe that each time the line touches a convex polytope, it only touches its surface twice. The exit point is the entrance point for a new polytope, except at the end-point. Thus, the variable that we are interested in can be represented as

$$Z = |\mathcal{F} \cap G|,$$

where $|\cdot|$ represents the cardinality of a set. Obviously, $\mathbb{E}\, Z$ is bounded by $1 < \mathbb{E}\, Z < K$. We will give a tighter bound in the sequel.

Let $\mathcal{G}$ be the class of all lines of $\mathbb{R}^D$, and μ be the measure of $\mathcal{G}$. Following the notation introduced in [15], we denote the measure of $\mathcal{G}$ that meet a fixed bounded convex set $\mathcal{C}$ as $\mu(\mathcal{G}; \mathcal{G} \cap \mathcal{C} \neq \emptyset)$. Considering an *independent Poisson point process* on $\mathcal{G}$ intensity measure μ, let N be the number of lines intersecting $\mathcal{X}$. We emphasize that N is a finite number, so that one can label them independently $G_1, \ldots, G_N$. It follows that $G_n, n = 1, \ldots, N$ are *i.i.d.*. Given a fixed classifier f, we have

$$\begin{aligned} \mathbb{E} \sum_{n=1}^{N} |\mathcal{F} \cap G_n| &= \mathbb{E} \sum_{n=1}^{N} \left[P(N=n) \sum_{i=1}^{n} |\mathcal{F} \cap G_i| \right] \\ &= \sum_{n=1}^{N} \left[P(N=n) \cdot n \cdot \mathbb{E} |\mathcal{F} \cap G_1| \right] \\ &= \mathbb{E}\, N \cdot (\mathbb{E}\, Z). \end{aligned} \tag{5}$$

Remark that $G_1, \ldots, G_N$ follow the Possion point process, we have $\mathbb{E}\, N = \mu(\mathcal{G}; \mathcal{G} \cap \mathcal{X} \neq \emptyset)$. Therefore we can rewrite (5) as,

$$\mathbb{E}\, Z = \frac{\mathbb{E} \sum_{n=1}^{N} |\mathcal{F} \cap G_n|}{\mu(\mathcal{G}; \mathcal{G} \cap \mathcal{X} \neq \emptyset)}. \tag{6}$$

Next, we compute $\mathbb{E}\sum_{n=1}^{N}|\mathcal{F}\cap G_n|$. Let $M = |\mathcal{F}|$. Due to the convexity of $\mathcal{X}_k$, any given line can hit a facet no more than once. Therefore, we have

$$\begin{aligned}\mathbb{E}\sum_{n=1}^{N}|\mathcal{F}\cap G_n| &= \mathbb{E}\sum_{n=1}^{N}\sum_{m=1}^{M}|F_m\cap G_n|\\ &= \sum_{m=1}^{M}\mathbb{E}\Big|\Big\{n\in\{1,\dots,N\}|F_m\cap G_n\neq\emptyset\Big\}\Big|\\ &= \sum_{m=1}^{M}\mu(\mathcal{G};\mathcal{G}\cap F_m\neq\emptyset). \qquad (7)\end{aligned}$$

By substituting (7) into (6) we obtain

$$\mathbb{E}\,Z = \frac{\sum_{m=1}^{M}\mu(\mathcal{G};\mathcal{G}\cap F_m\neq\emptyset)}{\mu(\mathcal{G};\mathcal{G}\cap\mathcal{X}\neq\emptyset)}. \qquad (8)$$

Assume that μ is translation invariant, by Cauchy-Crofton formula we can rewrite (8) as

$$\mathbb{E}\,Z = \frac{\sum_{m=1}^{M}A(F_m)}{A(\mathcal{X})}, \qquad (9)$$

where $A(\cdot)$ denotes the surface area[2]. Note, that the numerator of (9) depends on the shape of each polytope and relates to the training method of classifier. Thus, it is difficult to compute the exact value of $\mathbb{E}\,Z$. Nonetheless, we can bound the expectation by using the fact $A(\mathcal{X}) < \sum_{m=1}^{M}A(F_m) < A(\mathcal{X}) + \sqrt{2}(K-1)$ (see [1] for the upper bound). Remark that the surface area $A(\mathcal{X})$ of a unit hypercube is $2D$. We yield

$$1 < \mathbb{E}\,Z < 1 + \frac{\sqrt{2}(K-1)}{2D},$$

which concludes our proof. □

We remark that Theorem 3 implies a way to construct a robust classifier that resists evasion algorithm based on linear probing, e.g. by jointly minimizing (9) and the error function in the training procedure.

4 Algorithm for Approximating ϵ-IMAC

Based on theoretical results, we present an algorithm for deceiving the multi-class linear classifier by disguising the adversarial instance $\mathbf{x}^{\mathrm{A}}$ as other classes with approximately minimal cost, while issuing polynomially many queries in: the number of features, the range of feature, the number of classes and the number of iterations.

An outline of our searching approach is presented in Algorithms 1 to 3. We use a $K\times D$ matrix Ψ for storing ISMAC of K classes and an array C of length K for

[2] The surface area in $\mathbb{R}^D$ is the $(D-1)$-dimensional Lebesgue measure.

the corresponding adversarial cost of these instances. The scalar value W represents the maximal cost of all optimum instances. Additionally, we need a $K \times I$ matrix T for storing the searching path of optimum instances in each iteration. The k^{th} row of matrix Ψ is denoted as $\Psi[k,:]$. We consider Ψ, T, C, W as global variables so they are accessible in every scope. After initializing variables, our main routine `MLCEvading` (Algorithm 1 line 4) first invokes `MDSearch` (Algorithm 2) to search instances that is close to the starting point $\mathbf{x}^{\text{A}}$ in all classes and saves them to Ψ. Then it repetitively selects instances from Ψ as new starting points and searches instances with lower adversarial cost (Algorithm 3 line 6–7). The whole procedure iterates I times. Finally, we obtain $\Psi[k,:]$ as the approximation of ϵ-IMAC$(k, \mathbf{x}^{\text{A}})$.

We begin by describing `RBSearch` in Algorithm 3, a subroutine for searching instances near decision boundaries along dimension d. Essentially, given an instance $\mathbf{x}$, an upper bound u and a lower bound l, we perform a recursive binary search on the line segment $\{\mathbf{x} + \theta\boldsymbol{\delta}_d \,|\, l \leq \theta \leq u\}$ through $\mathbf{x}$. The effectiveness of this recursive algorithm relies on the fact that it is impossible to have $\mathbf{x}^{\text{u}}$ and $\mathbf{x}^{\text{l}}$ in the same class while $\mathbf{x}^{\text{m}}$ is in another class. In particular, if the line segment meets an exterior margin $\mathcal{M}_{+\epsilon}(k)$ and ϵ-IMAC$(k, \mathbf{x})$ is the intersection, then `RBSearch` finds an ϵ-IMAC. Otherwise, when the found instance $\mathbf{y}$ yields lower adversarial cost than instance in Ψ does, Algorithm 4 is invoked to update Ψ. The time complexity of `RBSearch` is $\mathcal{O}(\frac{u-l}{\epsilon})$.

We next describe Algorithm 2. Given $\mathbf{x}$ which is known as ISMAC$(k, \mathbf{x}^{\text{A}})$ and the current maximum cost W, the algorithm iterates $(D-1)$ times on $\mathcal{P}_{+\epsilon}(\mathcal{X}_{f(\mathbf{x})})$ for finding instances with cost lower than W. Additionally, we introduce two heuristics to prune unnecessary queries. First, the searched dimension in the previous iteration of $\mathbf{x}$ is omitted. Second, we restrict the upper and lower bound of the searching scope on each dimension. Specifically, knowing W and $a(\mathbf{x}, \mathbf{x}^{\text{A}}) = c$, we only allow `RBSearch` to find instance in $[x_d - \frac{W-c}{e_d}, x_d + \frac{W-c}{e_d}]$ since any instance lying out of this scope gives adversarial cost higher than W. This pruning is significant when we have obtained ISMAC for every class. Special attention must be paid to searched dimensions of $\mathbf{x}$ (see Algorithm 2 line 5–7). Namely, if d is a searched dimension before the $(i-1)^{\text{th}}$ iteration, then we relax the searching scope to $[x_d^{\text{A}} - \frac{W-c}{e_d}, x_d^{\text{A}} + \frac{W-c}{e_d}]$ so that no low-cost instances will be missed.

Algorithm 1. Query algorithm for evasion of multi-class linear classifiers

```
   $(\Psi, C) \leftarrow$ MLCEvading($\mathbf{x}^{\text{A}}, \mathbf{e}, D, L, U, K, I, \epsilon$):
1  for $k \leftarrow 1$ to $K$ do
2      $\Psi[k,:] \leftarrow \mathbf{0}$, $T[k,:] \leftarrow \mathbf{0}$, $C[k] \leftarrow +\infty$
3  $C[1] \leftarrow 0$
4  MDSearch($\mathbf{x}^{\text{A}}, \mathbf{x}^{\text{A}}, \mathbf{e}, 1, 0, D, L, U, 1, \epsilon$)
5  for $i \leftarrow 2$ to $I$ do
6      for $k \leftarrow 2$ to $K$ do
7          MDSearch($\Psi[k,:], \mathbf{x}^{\text{A}}, \mathbf{e}, k, C[k], D, L, U, i, \epsilon$)
```

Algorithm 2. Multi-dimensional search from ISMAC$(k, \mathbf{x}^{\mathrm{A}})$

```
MDSearch($\mathbf{x}, \mathbf{x}^{\mathrm{A}}, \mathbf{e}, k, c, D, L, U, i, \epsilon$):
for $d \leftarrow 1$ to $D$ do
    if $d \neq T[k, i-1]$ then
        $\delta \leftarrow \frac{W-c}{e_d}$
        $u = \min\{U, x_d + \delta\}$, $l = \max\{L, x_d - \delta\}$
        if $d \in \{T[k,1], \ldots, T[k, i-2]\}$ then
            if $x_d > x_d^{\mathrm{A}}$ then $l = \max\{L, x_d^{\mathrm{A}} - \delta\}$
            else $u = \min\{U, x_d^{\mathrm{A}} + \delta\}$
        $\mathbf{x}^{\mathrm{u}} \leftarrow \mathbf{x}$, $\mathbf{x}^{\mathrm{l}} \leftarrow \mathbf{x}$
        $x_d^{\mathrm{u}} \leftarrow u$, $x_d^{\mathrm{l}} \leftarrow l$
        if $f(\mathbf{x}^{\mathrm{u}}) \neq k$ then RBSearch($x_d, u, \mathbf{x}, d, i, \epsilon$)
        if $f(\mathbf{x}^{\mathrm{l}}) \neq k$ then RBSearch($l, x_d, \mathbf{x}, d, i, \epsilon$)
```

Algorithm 3. Recursive binary search on dimension d

```
RBSearch($l, u, \mathbf{x}, d, i, \epsilon$):
$\mathbf{x}^* \leftarrow \mathbf{x}$
if $u - l < \epsilon$ then
    $x_d^* \leftarrow u$
    $k \leftarrow f(\mathbf{x}^*)$, $c \leftarrow a(\mathbf{x}^*)$
    if $c < C[k]$ then Update($\mathbf{x}^*, k, c, d, i$)
$\mathbf{x}^{\mathrm{u}} \leftarrow \mathbf{x}$, $\mathbf{x}^{\mathrm{l}} \leftarrow \mathbf{x}$, $\mathbf{x}^{\mathrm{m}} \leftarrow \mathbf{x}$
$x_d^{\mathrm{u}} \leftarrow u$, $x_d^{\mathrm{l}} \leftarrow l$, $x_d^{\mathrm{m}} \leftarrow \frac{u+l}{2}$
if $f(\mathbf{x}^{\mathrm{m}}) = f(\mathbf{x}^{\mathrm{l}})$ then
    RBSearch($m, u, \mathbf{x}, d, i, \epsilon$)
else if $f(\mathbf{x}^{\mathrm{m}}) = f(\mathbf{x}^{\mathrm{u}})$ then
    RBSearch($l, m, \mathbf{x}, d, i, \epsilon$)
else
    RBSearch($l, m, \mathbf{x}, d, i, \epsilon$)
    RBSearch($m, u, \mathbf{x}, d, i, \epsilon$)
```

Algorithm 4. Update ISMAC$(k, \mathbf{x}^{\mathrm{A}})$

```
$(\Psi, C, T, W) \leftarrow$Update($\mathbf{x}^*, k, c, d, i$):
$\Psi[k, :] \leftarrow \mathbf{x}^*$
$C[k] \leftarrow c$
$T[k, i] \leftarrow d$
$W \leftarrow \max\{C[1], \ldots, C[K]\}$
```

Theorem 4. *The asymptotic time complexity of our algorithm is* $\mathcal{O}(\frac{U-L}{\epsilon}DKI)$.

Proof. Follows from the correctness of the algorithm and the fact that the time complexity of `RBSearch` is $\mathcal{O}(\frac{u-l}{\epsilon})$. □

5 Experiments

We demonstrate the algorithm[3] on two real-world data sets, the 20-newsgroups[4] and the 10-Japanese female face[5]. On the newsgroups data set, the task of the adversary is to evade a text classifier by disguising a commercial spam as a message in other topics. On the face data set, the task of adversary is to deceive the classifier by disguising a suspect's face as an innocent. We employ LIBLINEAR [6] package to build target multi-class linear classifiers, which return labels of queried instances. The cost coefficients are set to $e_1 = \cdots = e_D = 1$ for both tasks. For the groundtruth solution, we directly solve the optimization problem with linear constraints (3) and (4) by using the models' parameters. We then measure the average empirical ϵ for $(K-1)$ classes, which is defined as $\widehat{\epsilon} = \frac{1}{K-1}\sum_{k\neq f(\mathbf{x}^{\mathrm{A}})}\left[\frac{C[k]}{\mathrm{MAC}(k,\mathbf{x}^{\mathrm{A}})} - 1\right]$, where $C[k]$ is the adversarial cost of disguised instance of class k. Evidently, small $\widehat{\epsilon}$ indicates better approximation of IMAC.

5.1 Spam Disguising

The training data used to configure the newsletter classifier consists of $7,505$ documents, which are partitioned evenly across 20 different newsgroups. Each document is represented as a $61,188$-dimensional vector, where each component is the number of occurrences of a word. The accuracy of the classifier on training data is 100% for every class. We set the category "misc.forsale" as the adversarial class. That is, given a random document in "misc.forsale", the adversary attempts to disguise this document as from other category; e.g. "rec.sport.baseball". Parameters of the algorithm are $K = 20, L = 0, U = 100, I = 10, \epsilon = 1$. The adversary is restricted to query at most $10,000$ times. The adversarial cost of each class is depicted in Fig. 1 (left).

5.2 Face Camouflage

The training data contains 210 gray-scaled images of 7 facial expressions (each with 3 images) posed by 10 Japanese female subjects. Each image is represented by a 100-dimensional vector using principal components. The accuracy of the classifier on training data is 100% for every class. We randomly pick a subject as an imaginary suspect. Given a face image of the suspect, the adversary camouflage this face to make it be classified as other subjects. Parameters of the algorithm are $K = 10, L = -10^5, U =$

[3] A Matlab implementation is available at
http://home.in.tum.de/~xiaoh/pakdd2012-code.zip

[4] http://people.csail.mit.edu/jrennie/20Newsgroups/

[5] http://www.kasrl.org/jaffe.html

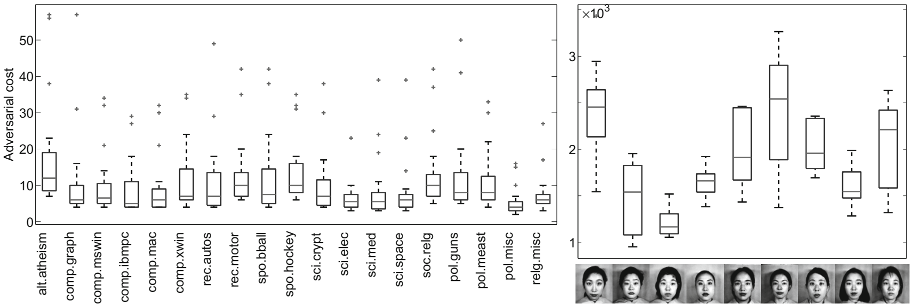


Fig. 1. Box plots for adversarial cost of disguised instance of each class. **(Left)** On the 20-newsgroups data set, we consider "misc.forsale" as the adversarial class. Note, that feature values of the instance are non-negative integers as they represent the number of words in the document. Therefore, the adversarial cost can be interpreted as the number of modified words in the disguised document comparing to the original document from "misc.forsale". The value of $\widehat{\epsilon}$ for 19 classes is 0.79. **(Right)** On the 10-Japanese female faces data set, we randomly select a subject as the suspect. The box plot shows that the adversarial cost of camouflage suspicious faces as other subjects. The value of $\widehat{\epsilon}$ for 9 classes is 0.51. A more illustrative result is depicted in Fig. 2.

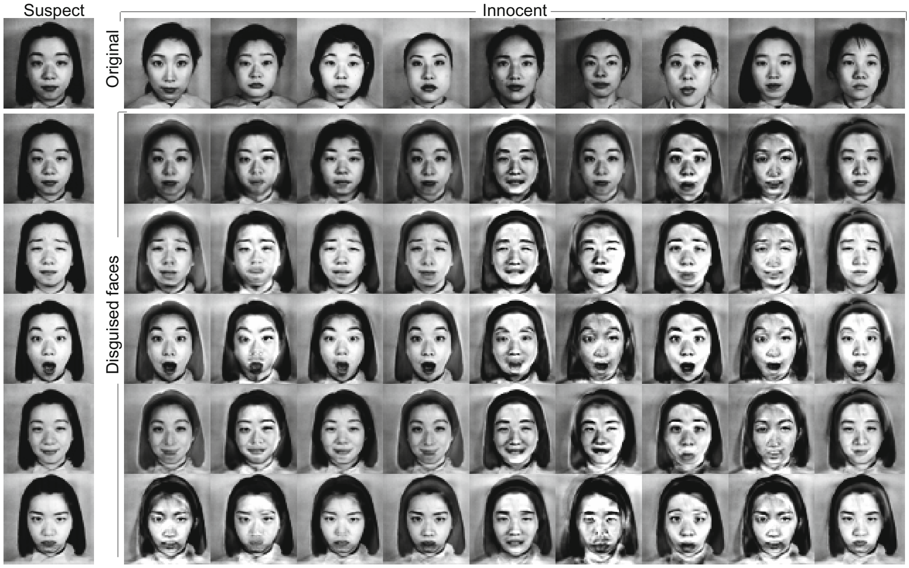


Fig. 2. Disguised faces given by our algorithm to defeat a multi-class face recognition system. The original faces (with neutral expression) of 10 females are depicted in the first row, where the left most one is the imaginary suspect and the remaining 9 people are innocents. From the second row to sixth row, faces of the suspect with different facial expressions are fed to the algorithm (see the first column). The output disguised faces from the algorithm are visualized in the right hand image matrix. Each row corresponds to disguised faces of the input suspicious face on the left. Each column corresponds to an innocent.

$10^5, I = 10, \epsilon = 1$. The adversary is restricted to query at most $10,000$ times. The adversarial cost of each class is depicted in Fig. 1 (right). Moreover, we visualize disguised faces in Fig. 2. Observe that many disguised faces are similar to the suspect's face by humans interpretation, yet they are deceptive for the classifier. This visualization directly demonstrates the effectiveness of our algorithm.

It has not escaped our notice that an experienced adversary with certain domain knowledge can reduce the number of queries by careful selecting cost function and employing heuristics. Nonetheless, the goal of this paper is not to design real attacks but rather examine the correctness and effectiveness of our algorithm so as to understand vulnerabilities of classifiers.

6 Conclusions

Adversary and classifier are *Yin* and *Yang* of information security. We believe that understanding the vulnerability of classifiers is the only way to develop resistant classifiers in the future. In this paper, we showed that multi-class linear classifiers are vulnerable to the evasion attack and presented an algorithm for disguising the adversarial instance. Future work includes generalizing the evasion attack problem to the family of general multi-class classifier with nonlinear decision boundaries.

References

1. Ball, K.: Cube slicing in $\mathbb{R}^n$. Proc. American Mathematical Society 97(3), 465–473 (1986)
2. Barbara, D., Jajodia, S.: Applications of data mining in computer security. Springer (2002)
3. Bratko, A., Filipič, B., Cormack, G., Lynam, T., Zupan, B.: Spam filtering using statistical data compression models. JMLR 7, 2673–2698 (2006)
4. Crammer, K., Singer, Y.: On the learnability and design of output codes for multiclass problems. Machine Learning 47(2), 201–233 (2002)
5. Dalvi, N., Domingos, P., et al.: Adversarial classification. In: Proc. 10th SIGKDD, pp. 99–108. ACM (2004)
6. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. JMLR 9, 1871–1874 (2008)
7. Fumera, G., Pillai, I., Roli, F.: Spam filtering based on the analysis of text information embedded into images. JMLR 7, 2699–2720 (2006)
8. Grünbaum, B.: Convex polytopes, vol. 221. Springer (2003)
9. Keerthi, S., Sundararajan, S., Chang, K., Hsieh, C., Lin, C.: A sequential dual method for large scale multi-class linear svms. In: Proc. 14th SIGKDD, pp. 408–416. ACM (2008)
10. Lowd, D., Meek, C.: Adversarial learning. In: Proc. 11th SIGKDD, pp. 641–647. ACM (2005)
11. Lowd, D., Meek, C.: Good word attacks on statistical spam filters. In: Proc. 2nd Conference on Email and Anti-Spam, pp. 125–132 (2005)
12. Maloof, M.: Machine learning and data mining for computer security: methods and applications. Springer (2006)
13. Nelson, B., Rubinstein, B.I.P., Huang, L., Joseph, A.D., Hon Lau, S., Lee, S., Rao, S., Tran, A., Tygar, J.D.: Near-optimal evasion of convex-inducing classifiers. In: Proc. 13th AISTATS (2010)
14. Rockafellar, R.: Convex analysis, vol. 28. Princeton Univ. Pr. (1997)
15. Santaló, L.: Integral geometry and geometric probability. Cambridge Univ. Pr. (2004)

Foundation of Mining Class-Imbalanced Data

Da Kuang, Charles X. Ling, and Jun Du

Department of Computer Science
The University of Western Ontario, London, Ontario, Canada N6A 5B7
{dkuang,cling,jdu42}@csd.uwo.ca

Abstract. Mining class-imbalanced data is a common yet challenging problem in data mining and machine learning. When the class is imbalanced, the error rate of the rare class is usually much higher than that of the majority class. How many samples do we need in order to bound the error of the rare class (and the majority class)? If the misclassification cost of the class is known, can the cost-weighted error be bounded as well? In this paper, we attempt to answer those questions with PAC-learning. We derive several upper bounds on the sample size that guarantee the error on a particular class (the rare and majority class) and the cost-weighted error, with the consistent and agnostic learners. Similar to the upper bounds in traditional PAC learning, our upper bounds are quite loose. In order to make them more practical, we empirically study the pattern observed in our upper bounds. From the empirical results we obtain some interesting implications for data mining in real-world applications. As far as we know, this is the first work providing theoretical bounds and the corresponding practical implications for mining class-imbalanced data with unequal cost.

1 Introduction

In data mining, datasets are often imbalanced (or class imbalanced); that is, the number of examples of one class (the rare class) is much smaller than the number of the other class (the majority class).[1]

This problem happens often in real-world applications of data mining. For example, in medical diagnosis of a certain type of cancer, usually only a small number of people being diagnosed actually have the cancer; the rest do not. If the cancer is regarded as the positive class, and non-cancer (healthy) as negative, then the positive examples may only occur 5% in the whole dataset collected. Besides, the number of fraudulent actions is much smaller than that of normal transactions in credit card usage data. When a classifier is trained on such an imbalanced dataset, it often shows a strong bias toward the majority class, since the goal of many standard learning algorithms is to minimize the overall prediction error rate. Thus, by simply predicting every example as the majority class, the classifier can still achieve a very low error rate on a class-imbalanced dataset with, for example, 2% rare class.

When mining the class-imbalanced data, do we always get poor performance (e.g., 100% error) on the rare class? Can the error of the rare class (as well as the majority class) be bounded? If so, is the bound sensitive to the class imbalance ratio? Although

[1] In this paper, we only study binary classification.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 219–230, 2012.

the issue of class imbalance has been received extensive studies [9,3,2,7,4,5], as far as we know, no previous works have been done to answer those questions.

In fact, PAC learning (Probably Approximately Correct Learning) [8,6] is an appropriate model to study the bounds for classification performance. The traditional PAC learning model studies the learnability of the general concept for certain kinds of learners (such as consistent learner and agnostic learner), and answers the question that how many examples would be sufficient to guarantee a low total error rate. However, previous works [9] point out that accuracy or total error rate are inappropriate to evaluate the classification performance when class is imbalanced, since such metrics overly emphasize the majority class and neglect the rare class which is usually more important in real-world applications. Thus, when class is imbalanced, better measures are desired. In our paper, we will use error rate on the rare (and majority) class and *cost-weighted error*[2] to evaluate the classification performance on class-imbalanced data. The error rate on the rare (and majority) class can reflect how well the rare (and majority) class is learned. If the misclassification cost of the class is known, we can adopt another common measure (cost-weighted error) to deal with imbalanced data. By weighting the error rate on each class by its associated cost, we will get higher penalty for the error on the rare class (usually the more important class).

In our paper, we attempt to use the PAC-learning model to study, when class is imbalanced, how many sampled examples needed to guarantee a low error on a particular class (the rare class or majority class) and a low cost-weighted error respectively. A bound on cost-weighted error is necessary since it would naturally "suppress" errors on the rare class. We theoretically derive several upper bounds for both consistent learner and agnostic learner. Similar to the upper bounds in traditional PAC learning, the bounds we derive are generally quite loose, but they do provide a theoretical guarantee on the classification performance when class-imbalanced data is learned. Due to the loose bounds, to make our work more practical, we also empirically study how class imbalance affects the performance by using a specific learner. From our experimental results, some interesting implications can be found. The results in this paper can provide some theoretical foundations for mining the class-imbalanced data in the real world.

The rest of the paper is organized as follows. We theoretically derive several upper bounds on the sample complexity for both consistent learner and agnostic learner. Then we empirically explore how class imbalance affects the classification performance by using a specific learner. Finally, we draw the conclusions and address our future work.

2 Upper Bounds

In this section, we take advantage of PAC-learning theory to study the sample complexity when learning from the class-imbalanced data. Instead of bounding the total error rate, we focus on the error rate on a particular class (rare class or majority class) and the cost-weighted error.

[2] We will define it in the next section.

2.1 Error Rate on a Particular Class

First of all, we introduce some notations for readers' convenience. We assume that the examples in training set T are drawn randomly and independently from a fixed but unknown class-imbalanced distribution D. We denote p $(0 < p < 0.5)$ as the proportion of the rare class (the positive class) in D. For the class-imbalanced training set, p can be very small (such as 0.001). The number of total training examples is denoted as m and the number of positive and negative training examples are denoted as m_+ and m_- respectively. For any hypothesis h from the hypothesis space H, we denote $e_D(h)$, $e_{D+}(h)$, and $e_{D-}(h)$ as the total, the positive, and the negative generalization error, respectively, of h, and we also denote $e_T(h)$, $e_{T+}(h)$, and $e_{T-}(h)$ as the total, the positive, and the negative training error, respectively, of h.

Given ε $(0 < \varepsilon < 1)$ and δ $(0 < \delta < 1)$, the traditional PAC learning provides upper bounds on the total number of training examples needed to guarantee $e_D(h) < \varepsilon$ with probability at least $1 - \delta$. However, it guarantees nothing about the positive error $e_{D+}(h)$ for the imbalanced datasets. As we discussed before, the majority classifier would predict every example as negative, resulting in a 100% error rate on the positive (rare) examples. To have a lower positive error, the learner should observe more positive examples. Thus, in this subsection, we study the upper bounds of the examples on a particular class (say positive class here) needed to guarantee, with probability at least $1 - \delta$, $e_{D+}(h) < \varepsilon_+$, given any ε_+ $(0 < \varepsilon_+ < 1)$.

We first present a simple relation between the total error and the positive error as well as the negative error, and will use it to derive some upper bounds.

Theorem 1. *Given any ε_+ $(0 < \varepsilon_+ < 1)$ and the positive class proportion p $(0 < p < 0.5)$ according to distribution D and target function C, for any hypothesis h, if $e_D(h) < \varepsilon_+ \times p$, then $e_{D+}(h) < \varepsilon_+$.*

Proof. To prove this, we simply observe that,

$$e_D(h) = e_{D+}(h) \times p + e_{D-}(h) \times (1-p) \geq e_{D+}(h) \times p.$$

Thus,

$$e_{D+}(h) \leq \frac{e_D(h)}{p}.$$

Therefore, if $e_D(h) < \varepsilon_+ \times p$, $e_{D+}(h) < \varepsilon_+$.

Following the same direction, we can also derive a similar result for the error on negative class $e_{D-}(h)$. That is, given ε_- $(0 < \varepsilon_- < 1)$, if $e_D(h) < \varepsilon_- \times (1-p)$, then $e_{D-}(h) < \varepsilon_-$.

Theorem 1 simply tells us, as long as the total error is small enough, a desired positive error (as well as negative error) can always be guaranteed. Based on Theorem 1, we can "reuse" the upper bounds in the traditional PAC learning model and adapt them to be the upper bounds of a particular class in the class-imbalanced datasets. We first consider consistent learner in the next subsection.

Consistent Learner. We consider *consistent learner L* using hypothesis space H by assuming that the target concept c is representable by H ($c \in H$). Consistent learner always makes correct prediction on the training examples. Let us assume that $UB(\varepsilon,\delta)$ is an upper bound on the sample size in the traditional PAC-learning, which means that, given ε ($0 < \varepsilon < 1$) and δ ($0 < \delta < 1$), if the total number of training examples $m \geq UB(\varepsilon,\delta)$, a consistent learner will produce a hypothesis h such that with the probability at least $(1-\delta)$, $e_D(h) \leq \varepsilon$. The following theorem shows that we can adapt any upper bound in the traditional PAC-learning to the bounds that guarantee a low error on the positive class and negative class respectively.

For any upper bound of a consistent PAC learner $UB(\varepsilon,\delta)$, we can always replace ε in $UB(\varepsilon,\delta)$ with $\varepsilon_+ \times p$ or $\varepsilon_- \times (1-p)$, and consequently obtain a upper bound to guarantee the error rate on that particular class.

Theorem 2. *Given $0 < \varepsilon_+ < 1$, if the number of positive examples*

$$m_+ \geq UB(\varepsilon_+ \times p, \delta) \times p,$$

then with probability at least $1-\delta$, the consistent learner will output a hypothesis h having $e_{D+}(h) \leq \varepsilon_+$.

Proof. By the definition of the upper bound for the sample complexity, given $0 < \varepsilon < 1$, $0 < \delta < 1$, if $m \geq UB(\varepsilon,\delta)$, with probability at least $1-\delta$ any consistent learner will output a hypothesis h having $e_D(h) \leq \varepsilon$.

Here, we simply substitute ε in $UB(\varepsilon,\delta)$ with $\varepsilon_+ \times p$, which is still within (0, 1). Consequently, we obtain that if $m \geq UB(\varepsilon_+ \times p, \delta)$, with probability at least $1-\delta$ any consistent learner will output a hypothesis h having $e_D(h) \leq \varepsilon_+ \times p$. According to Theorem 1, we get $e_{D+}(h) < \varepsilon_+$.

Also, $m = \frac{m_+}{p}$, thus we know, $m \geq UB(\varepsilon_+ \times p, \delta)$ equals to

$$m_+ \geq UB(\varepsilon_+ \times p, \delta) \times p.$$

Thus, the theorem is proved.

By using the similar proof to Theorem 2, we can also derive the upper bound for the negative class. Given $0 < \varepsilon_- < 1$, if the number of negative examples $m_- \geq UB(\varepsilon_- \times (1-p), \delta) \times (1-p)$, then, with probability at least $1-\delta$, the consistent learner will output a hypothesis h having $e_{D_-}(h) \leq \varepsilon_-$.

The two upper bounds above can be adapted to any traditional upper bound of consistent learners. For instance, it is well known that any consistent learner using finite hypothesis space H has an upper bound $\frac{1}{\varepsilon} \times (ln|H| + ln\frac{1}{\delta})$ [6]. Thus, by applying our new upper bounds, we obtain the following corollary.

Corollary 1. *For any consistent learner using finite hypothesis space H, the upper bound on the number of positive sample for $e_{D+}(h) \leq \varepsilon_+$ is*

$$m_+ \geq \frac{1}{\varepsilon_+}(ln|H| + ln\frac{1}{\delta}),$$

and the upper bound on the number of negative sample for $e_{D-}(h) \leq \varepsilon_{-}$ *is*

$$m_{-} \geq \frac{1}{\varepsilon_{-}}(ln|H| + ln\frac{1}{\delta}).$$

From Corollary 1, we can discover that when the consistent learner uses *finite* hypothesis space, the upper bound of sample size on a particular class is directly related to the desired error rate (ε_{+} or ε_{-}) on the class, and the class imbalance ratio p does not affect the upper bound. This indicates that, for consistent learner, no matter how class-imbalanced the data is (how small p is), as soon as we sample sufficient examples in a class, we can always achieve the desired error rate on that class.

Agnostic Learner. In this subsection, we consider *agnostic learner L* using finite hypothesis space H, which makes *no* assumption about whether or not the target concept c is representable by H. Agnostic learner simply finds the hypothesis with the minimum (probably non-zero) training error. Given an arbitrary small ε_{+}, we can not ensure $e_{D+}(h) \leq \varepsilon_{+}$, since very likely $e_{T+}(h) > \varepsilon_{+}$. Hence, we guarantee $e_{D+}(h) \leq e_{T+}(h) + \varepsilon$ to happen with probability higher than $1 - \delta$, for such h with the minimum training error. To prove the upper bound for agnostic learner, we adapt the original proof for agnostic learner in [6]. The original proof regards drawing m examples from the distribution D as m independent Bernoulli trials, but in our proof, we only treat drawing m_{+} examples from the positive class as m_{+} Bernoulli trials.

Theorem 3. *Given* ε_{+} *(*$0 < \varepsilon_{+} < 1$*), any* δ *(*$0 < \delta < 1$*), if the number of positive examples observed*

$$m_{+} > \frac{1}{2\varepsilon_{+}^{2}}(ln|H| + ln\frac{1}{\delta}),$$

then with probability at least $1 - \delta$*, the agnostic learner will output a hypothesis h, such that* $e_{D+}(h) \leq e_{T+}(h) + \varepsilon_{+}$

Proof. For any h, we consider $e_{D+}(h)$ as the true probability that h will misclassify a randomly drawn positive example. $e_{T+}(h)$ is an observed frequency of misclassification over the given m_{+} positive training examples. Since the entire training examples are drawn identically and independently, drawing and predicting positive training examples are also identical and independent. Thus, we can treat drawing and predicting m_{+} positive training examples as m_{+} independent Bernoulli trials.

Therefore, according to Hoeffding bounds, we can have,

$$Pr[e_{D+}(h) > e_{T+}(h) + \varepsilon] \leq e^{-2m_{+}\varepsilon^{2}}.$$

According to the inequation above, we can derive,

$$Pr[(\exists h \in H)(e_{D+}(h) > e_{T+}(h) + \varepsilon)] \leq |H|e^{-2m_{+}\varepsilon^{2}}.$$

This formula tells us that the probability that there exists one bad hypothesis h making $e_{D+}(h) > e_{T+}(h) + \varepsilon$ is bounded by $|H|e^{-2m_{+}\varepsilon^{2}}$. If we let $|H|e^{-2m_{+}\varepsilon^{2}}$ be less than δ, then for any hypothesis including the outputted hypothesis h in H, $e_{D+}(h) - e_{T+}(h) \leq \varepsilon$ will hold true with the probability at least $1 - \delta$. So, solving for m_+ in the inequation $|H|e^{-2m_+\varepsilon^2} < \delta$, we obtain

$$m_+ > \frac{1}{2\varepsilon_+^2}(ln|H| + ln\frac{1}{\delta}).$$

Thus, the theorem is proved.

In fact, by using the similar procedure, we can also prove the upper bound for the number of negative examples m_- when using agnostic learner: $\frac{1}{2\varepsilon_-^2}(ln|H| + ln\frac{1}{\delta})$.

We can observe a similar pattern here. The upper bounds for the agnostic learner are also not affected by the class imbalance ratio p.

From the upper bound of either consistent learner or agnostic learner we derived, we learned that when the amount of examples on a class is enough, class imbalance does not take any effect. This discovery actually refutes a common misconception that we need more examples just because of the more imbalanced class ratio. We can see, the class imbalance is in fact a data insufficiency problem, which was also observed empirically in [4]. Here, we further confirm it with our theoretical analysis.

In this subsection, we derive a new relation (Theorem 1) between the positive error and the total error, and use it to derive a general upper bound (Theorem 2) which can be applied to any traditional PAC upper bound for consistent learner. We also extend the existing proof of agnostic learner to derive a upper bound on a particular class for agnostic learner. Although the proof of the theorems above may seem straightforward, no previous work explicitly states the same conclusion from the theoretical perspective.

It should be noted that although the agnostic learner outputs the hypothesis with the minimum (total) training error, it is possible that the outputted hypothesis has 100% error rate on the positive class in the training set. In this case, the guaranteed small difference ε_+ between the true positive error and the training positive error can still result in 100% true error rate on the positive class. If the positive errors are more costly than the negative errors, it is more reasonable to assign higher cost for misclassifying positive examples, and let the agnostic learner minimize the cost-weighted training error instead of the flat training error. In the following part, we will introduce misclassification cost to our error bounds.

2.2 Cost-Weighted Error

In this subsection, we take misclassification cost into consideration. We assume that the misclassification cost of the class is known, and the cost of a positive error (rare class) is higher than (at least equals) the cost of a negative error. We use C_{FN} and C_{FP} to represent the cost of misclassifying a positive example and a negative example, respectively.[3] And we denote r as the cost ratio, $\frac{C_{FN}}{C_{FP}}$ $(r \geq 1)$. Here we define a new type of error, named *cost-weighted error*.

[3] We assume the cost of correctly predicting a positive example and a negative example is 0, meaning that $C_{TP} = 0$ and $C_{TN} = 0$.

Definition 1 (Cost-Weighted Error). *Given the cost ratio r, the class ratio p, e_{D+} as the positive error on D, e_{D-} as the negative error on D, the cost-weighted error on D can be defined as,*

$$c_D(h) = \frac{rpe_{D+} + (1-p)e_{D-}}{rp + (1-p)}.$$

By the same definition, we can also define the cost-weighted error on the training set T as $c_T(h) = \frac{rpe_{T+}+(1-p)e_{T-}}{rp+(1-p)}$. The weight of the error on a class is determined by its class ratio and misclassification cost. The rp is the weight for the positive class and $1-p$ is the weight for the negative class. In our definition for the cost-weighted error, we use the normalized weight.

In the following part, we study the upper bounds for the examples needed to guarantee a low cost-weighted error on D. We give a non-trivial proof for the upper bounds of consistent learner, and the proof for the upper bound of agnostic learner is omitted due to its similarity to that of the consistent learner (but only with finite hypothesis space).

Consistent Learner. To derive a relatively tight upper bound of sample size for cost-weighted error, we first introduce a property. That is, there exist many combinations of positive error e_{D+} and negative error e_{D-} that can make the same cost-weighted error value. For example, given $rp = 0.4$, if $e_{D+} = 0.1$ and $e_{D-} = 0.2$, c_D will be 0.16, while $e_{D+} = 0.25$ and $e_{D-} = 0.1$ can also produce the same cost-weighted error. We can let the upper bound to be the least required sample size among all the combinations of positive error and negative error that can make the desired cost-weighted error.

Theorem 4. *Given ε ($0 < \varepsilon < 1$), any δ ($0 < \delta < 1$), the cost ratio r ($r \geq 1$) and the positive proportion p ($0 < p < 0.5$) according to the distribution D, if the total number of examples observed*

$$m \geq \frac{1+r}{\varepsilon(rp+(1-p))}(ln|H| + ln\frac{1}{\delta}),$$

then, with probability at least $1-\delta$, the consistent learner will output a hypothesis h such that the cost-weighted error $c_D(h) \leq \varepsilon$.

Proof. In order to make $c_D(h) \leq \varepsilon$, we should ensure,

$$\frac{rpe_{D+} + (1-p)e_{D-}}{rp + (1-p)} \leq \varepsilon. \qquad (1)$$

Here, we let $X = \frac{rp}{rp+(1-p)}$, thus $1-X = \frac{(1-p)}{rp+(1-p)}$. Accordingly, Formula (1) can be transformed into $Xe_{D+} + (1-X)e_{D-} \leq \varepsilon$. To guarantee it, we should make sure,

$$e_{D-} \leq \frac{\varepsilon - Xe_{D+}}{1-X}.$$

According to Corollary 1, if we observe,

$$m_- \geq \frac{1}{\frac{\varepsilon - Xe_{D+}}{1-X}}(ln|H| + ln\frac{1}{\delta}), \qquad (2)$$

we can also ensure $e_{D-}(h) \leq \frac{\varepsilon - Xe_{D+}}{1-X}$ with probability at least $1-\delta$ to happen. Besides, in order to have e_{D+} on positive class, we also need to observe,

$$m_+ \geq \frac{1}{e_{D+}}(ln|H| + ln\frac{1}{\delta}). \tag{3}$$

To guarantee Formula (2) and (3), we need to sample at least m examples such that $m = MAX(\frac{m_+}{p}, \frac{m_-}{1-p})$. Thus,

$$m \geq MAX(\frac{1}{e_{D+} \times p}, \frac{1}{\frac{\varepsilon - Xe_{D+}}{1-X} \times (1-p)})(ln|H| + ln\frac{1}{\delta})).$$

However, since e_{D+} is a variable, different e_{D+} will lead to different e_{D-}, and thus affect m. In order to have a tight upper bound for m, we only need,

$$m \geq \underset{0 \leq e_{D+} \leq \frac{\varepsilon}{X}}{MIN}(MAX(\frac{1}{e_{D+} \times p}, \frac{1}{\frac{\varepsilon - Xe_{D+}}{1-X} \times (1-p)})(ln|H| + ln\frac{1}{\delta})).$$

When $\frac{1}{e_{D+} \times p} > \frac{1}{\frac{\varepsilon - Xe_{D+}}{1-X} \times (1-p)}$, $MAX(\frac{1}{e_{D+} \times p}, \frac{1}{\frac{\varepsilon - Xe_{D+}}{1-X} \times (1-p)}) = \frac{1}{e_{D+} \times p}$, which is a decreasing function of e_{D+}, but when $\frac{1}{e_{D+} \times p} < \frac{1}{\frac{\varepsilon - Xe_{D+}}{1-X} \times (1-p)}$, it becomes an increasing function of e_{D+}. Thus, the minimum value of the function can be achieved when $\frac{1}{e_{D+} \times p} = \frac{1}{\frac{\varepsilon - Xe_{D+}}{1-X} \times (1-p)}$. By solving the equation, we obtain the minimum value for the function,

$$\frac{1}{\frac{\varepsilon(1-p)}{p+X-2Xp} \times p}(ln|H| + ln\frac{1}{\delta}).$$

If we recover X with $\frac{rp}{rp+(1-p)}$, then it can be transformed into $\frac{1+r}{\varepsilon(rp+(1-p))}(ln|H| + ln\frac{1}{\delta})$. Therefore, as long as,

$$m \geq \frac{1+r}{\varepsilon(rp+(1-p))}(ln|H| + ln\frac{1}{\delta}),$$

then with probability at least $1-\delta$, the consistent learner will output a hypothesis h such that $c_D(h) \leq \varepsilon$.

We can see that the upper bound of cost-weighted error for consistent learner is related to p and r. By performing a simple transformation, we can transform the above upper bound into $\frac{r+1}{\varepsilon((r-1)p+1)}(ln|H| + ln\frac{1}{\delta})$. It is known that $r \geq 1$, thus $r-1 \geq 0$. Therefore, as p decreases within (0, 0.5), the upper bound increases. It means that the more the class is imbalanced, the more examples we need to achieve a desired cost-weighted error. In this case, class imbalance actually affects the classification performance in terms of cost-weighted error. If we make another transformation to the upper bound, we can obtain, $\frac{1}{p\varepsilon} + \frac{2p-1}{\varepsilon(rp^2+(1-p)p)}(ln|H| + ln\frac{1}{\delta})$. Since $0 < p < 0.5$, $2p-1 < 0$. Thus, as r increases, the upper bound also increases. It shows that a higher cost ratio $\frac{C_{FN}}{C_{FP}}$ would require more examples for training. Intuitively speaking, when class is imbalanced, the cost-weighted error largely depends on the error on the rare class. As we have proved before, to achieve the same error on the rare class, we need the same amount of examples on the rare class, thus more class-imbalanced data requires more examples in total. Besides, higher cost on the rare class leads to higher cost-weighted error, thus to achieve the same cost-weighted error, we will also need more examples in total.

Agnostic Learner. As mentioned before, the hypothesis with the minimum training error produced by agnostic learner may still lead to 100% error rate on the rare class. Hence, instead of outputting the hypothesis with minimum training error, we redefine agnostic learner as the learner that outputs the hypothesis with the minimum cost-weighted error on the training set. Generally, with higher cost on positive errors, the agnostic learner is less likely to produce a hypothesis that misclassifies all the positive training examples. The following theorem demonstrates that, for agnostic learner, how many examples needed to guarantee a small difference of the cost-weighted errors between the distribution D and the training set T.

Theorem 5. *Given ε ($0 < \varepsilon < 1$), any δ ($0 < \delta < 1$), the cost ratio r ($r \geq 1$) and the positive proportion p ($0 < p < 0.5$) according to the distribution D, if the total number of examples observed*

$$m \geq \frac{r\sqrt{p}+\sqrt{1-p}}{2\varepsilon^2(rp+(1-p))}(ln|H|+ln\frac{1}{\delta}),$$

then, with probability at least $1-\delta$, the agnostic learner will output a hypothesis h such that $c_D(h) \leq c_T(h)+\varepsilon$.

The proof for Theorem 5 is very similar to that of Theorem 4, thus here we omit the detail of the proof. Furthermore, we can also extract the same patterns from the upper bound here as found for the upper bound in Theorem 4: more examples are required when the cost ratio increases or the class becomes more imbalanced.

To summarize, in this section we derive several upper bounds to guarantee the error rate on a particular class (rare class or majority class) as well as the cost-weighted error, for both consistent learner and agnostic learner. We found some interesting and useful patterns from those theoretical results: the upper bound for the error rate on a particular class is not affected by the class imbalance, while the upper bound for the cost-weighted error is sensitive to both the class imbalance and the cost ratio. Although those pattern may not be so surprising, as far as we know, no previous work theoretically proved it before. Such theoretical results would be more reliable than the results only based on the empirical observation.

Since the upper bounds we derive are closely related to the hypothesis space, which is often huge for many learning algorithms, they are generally very loose (It should be noted that in traditional PAC learning, the upper bounds are also very loose). In fact, when we practically use some specific learners, to achieve a desired error rate on a class or cost-weighted cost, usually the number of examples needed are much less than the theoretical upper bounds. Therefore, in the next section, we will empirically study the performance of a specific learner, to see how the class imbalance and cost ratio influence the classification performance.

3 Empirical Results with Specific Learner

In this section, we empirically explore the patterns found in the theoretical upper bounds. We hope to see, in practice, how the class imbalance and cost ratio affect the actual examples needed and whether the empirical results reflect our theories. Those empirical observations can be useful for practical data mining or machine learning with class-imbalanced data. In our following experiments, we will empirically study the performance of unpruned decision tree (consistent learner) on class-imbalanced datasets.[4]

3.1 Datasets and Settings

We choose unpruned decision tree for our empirical study, since it is a consistent learner in any case. It can be always consistent with the training data of any concept by building up a full large tree, if there are no conflicting examples with the same attribute values but different class labels. For the specific implementation, we use WEKA [10] and select *J48* with the pruning turned off and the parameter $MinNumObj = 1$.

We create one artificial dataset and select two real-world datasets. The artificial dataset we use is generated by a tree function with five relevant attributes, $A1 - A5$, and six leaves, as shown in Figure 1. To simulate the real-world dataset, we add another 11 irrelevant attributes. Therefore, with 16 binary attributes, we can generate $2^{16} = 65,536$ different examples, and label them with the target concept (28,672 positive and 36,864 negative). We also choose two UCI [1] real-world datasets (*Chess* and *Splice*). In order to make the unpruned decision tree with all the training examples, the conflicting examples (i.e., the examples with identical attribute values but different labels) are eliminated during the pre-process.

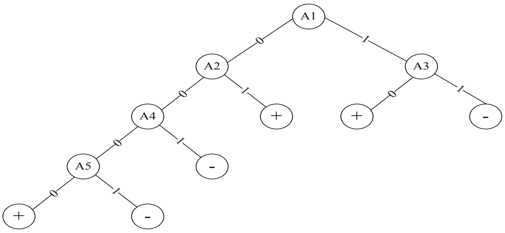


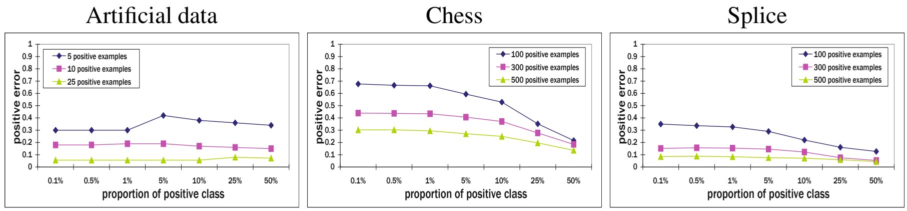
Fig. 1. Artificial tree function

3.2 Experimental Design and Results

To see how class imbalance affects the error rate on a particular class (here we choose positive class), we compare the positive error under different class ratios but with the same number of positive examples in the training set.

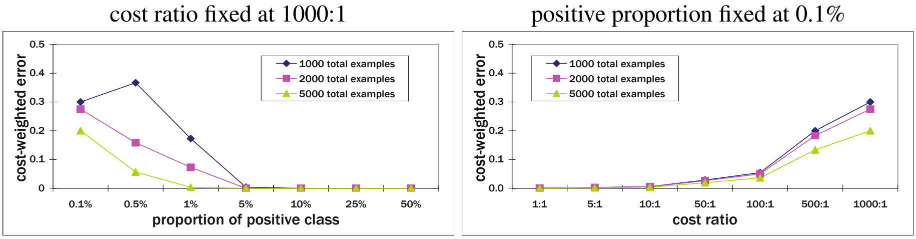
Specifically, we manually generate different data distributions with various class ratios where training set and test set are drawn. For example, to generate a data distribution with 10% positive proportion, we simply set the probability of drawing a positive

[4] Due to the limited pages, we only empirically study the consistent learner here.



Artificial data Chess Splice

Fig. 2. Positive error of unpruned decision tree on three datasets



cost ratio fixed at 1000:1 positive proportion fixed at 0.1%

Fig. 3. Cost-weighted error of unpruned decision tree on the artificial data

example to be 1/9 of the probability of drawing a negative example, and the probability of drawing examples within a class is uniform. According to the data distribution, we sample a training set until it contains a certain number of positive examples (we set three different numbers for each dataset), and train a unpruned decision tree on it. Then, we evaluate its performance (positive error and cost-weighted error) on another sampled test set from the same data distribution. Finally, we compare the performance under different data distributions (0.1%, 0.5%, 1%, 5%, 10%, 25%, 50%) to see how class imbalance ratio affects the performance of the unpruned decision tree. All the results are the average value over 10 independent runs.

Figure 2 presents the positive error on three datasets. The three curves in each subgraph represent three different numbers of positive examples in the training set. For the artificial dataset, since the concept is easy to learn, the number of positive examples chosen is smaller than that of the UCI datasets. We can see, generally the more the positive examples for training, the flatter the curve and the lower the positive error. It means, as we have more positive examples, class imbalance has less negative effect on the positive error in practice. The observation is actually consistent with Corollary 1.

To see how class imbalance influences the cost-weighted error, we compare the cost-weighted error under different class ratios with fixed cost ratio. To explore how cost ratio affects the cost-weighted error, we compare the cost-weighted error over different cost ratios with fixed class ratio. For this part, we only use the artificial dataset to show the results (see Figure 3). We can see, generally, as the class becomes more imbalanced or the cost ratio increases, the cost-weighted error goes higher. It is also consistent with our theory (Theorem 4).

We have to point out that, our experiment is not a verification of our derived theories. The actual amount of examples we used in our experiment is much smaller compared to the theoretical bounds. Despite of that, we still find that the empirical observations have similar patterns to our theoretical results. Thus, our theorems not only offer a theoretical guarantee, but also has some useful implications for real-world applications.

4 Conclusions

In this paper, we study the class imbalance issue from PAC-learning perspective. An important contribution of our work is that, we theoretically prove that the upper bound of the error rate on a particular class is not affected by the (imbalanced) class ratio. It actually refutes a common misconception that we need more examples just because of the more imbalanced class ratio. Besides the theoretical theorems, we also empirically explore the issue of the class imbalance. The empirical observations reflect the patterns we found in our theoretical upper bounds, which means our theories are still helpful for the practical study of class-imbalanced data.

Although intuitively our results might seem to be straightforward, few previous works have explicitly addressed these fundamental issues with PAC bounds for class-imbalanced data before. Our work actually confirms the practical intuition by theoretical proof and fills a gap in the established PAC learning theory. For imbalanced data issue, we do need such a theoretical guideline for practical study.

In our future work, we will study bounds for AUC, since it is another useful measure for the imbalanced data. Another common heuristic method to deal with imbalanced data is over-sampling and under-sampling. We will also study their bounds in the future.

References

1. Asuncion, A., Newman, D.: UCI machine learning repository (2007), http://www.ics.uci.edu/~mlearn/mlrepository.html
2. Carvalho, R., Freitas, A.: A hybrid decision tree/genetic algorithm method for data mining. Inf. Sci. 163(1-3), 13–35 (2004)
3. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)
4. Japkowicz, N.: Class imbalances: Are we focusing on the right issue? In: ICML-KDD 2003 Workshop: Learning from Imbalanced Data Sets (2003)
5. Klement, W., Wilk, S., Michalowski, W., Matwin, S.: Classifying Severely Imbalanced Data. In: Butz, C., Lingras, P. (eds.) Canadian AI 2011. LNCS, vol. 6657, pp. 258–264. Springer, Heidelberg (2011)
6. Mitchell, T.: Machine Learning. McGraw-Hill, New York (1997)
7. Ting, K.M.: The problem of small disjuncts: its remedy in decision trees. In: Proceeding of the Tenth Canadian Conference on Artificial Intelligence, pp. 91–97 (1994)
8. Valiant, L.G.: A theory of the learnable. Commun. ACM 27(11), 1134–1142 (1984)
9. Weiss, G.: Mining with rarity: a unifying framework. SIGKDD Explor. Newsl. 6(1), 7–19 (2004)
10. WEKA Machine Learning Project: Weka, http://www.cs.waikato.ac.nz/~ml/weka

Active Learning with c-Certainty

Eileen A. Ni and Charles X. Ling

Department of Computer Science
The University of Western Ontario
London, Ontario, Canada
{ani,cling}@csd.uwo.ca

Abstract. It is well known that the noise in labels deteriorates the performance of active learning. To reduce the noise, works on multiple oracles have been proposed. However, there is still no way to guarantee the label quality. In addition, most previous works assume that the noise level of oracles is evenly distributed or example-independent which may not be realistic. In this paper, we propose a novel active learning paradigm in which oracles can return both labels and confidences. Under this paradigm, we then propose a new and effective active learning strategy that can guarantee the quality of labels by querying multiple oracles. Furthermore, we remove the assumptions of the previous works mentioned above, and design a novel algorithm that is able to select the best oracles to query. Our empirical study shows that the new algorithm is robust, and it performs well with given different types of oracles. As far as we know, this is the first work that proposes this new active learning paradigm and an active learning algorithm in which label quality is guaranteed.

Keywords: Active learning, multiple oracles, noisy data.

1 Introduction

It is well known that the noise in labels deteriorates learning performance, especially for active learning, as most active learning strategies often select examples with noise on many natural learning problems [1]. To rule out the negative effects of the noisy labels, querying multiple oracles has been proposed in active learning [2,3,4]. This multiple-oracle strategy is reasonable and useful in improving label quality. For example, in paper reviewing, multiple reviewers (i.e., oracles or labelers) are requested to label a paper (as accepted, weak accepted, weak rejected or rejected), so that the final decision (i.e., label) can be more accurate.

However, there is still no way to guarantee the label quality in spite of the improvements obtained in previous works [3,4,5]. Furthermore, strong assumptions, such as even distribution of noise [3], and example-independent (fixed) noise level [4], have been made. These assumptions, in the paper reviewing example mentioned above, imply that all the reviewers are at the same level of expertise and have the same probability in making mistakes.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 231–242, 2012.

Obviously, the assumptions may be too strong and not realistic, as it is ubiquitous that label quality (or noise-level) is example-dependent in real-world data. In the paper reviewing example, the quality of a label given by a reviewer should depend heavily on how close the reviewer's research is to the topic of the paper. The closer it is, the higher quality the label has. Thus, it is necessary to study this learning problem further.

In this paper, we propose a novel active learning paradigm, under which oracles are assumed to return both labels and confidences. This assumption is reasonable in real-life applications. Taking paper reviewing as an example again, usually a reviewer is required to give not only a label (accept, weak accept, weak reject or reject) for a paper, but also his confidence (high, medium or low) for the labeling.

Under the paradigm, we propose a new active learning strategy, called *c-certainty learning*. C-certainty learning guarantees the label quality to be equal to or higher than a threshold c (c is the probability of correct labeling; see later) by querying oracles multiple times. In the paper reviewing example, with the labels and confidences given by reviewers (oracles), we can estimate the certainty of the label. If the certainty is too low (e.g., lower than a given c), another reviewer has to be sought to review the paper to improve the label quality.

Furthermore, instead of assuming noise level to be example-independent in the previous works, we allow it to be example-dependent. We design an algorithm that is able to select the *Best Multiple Oracles* to query (called *BMO*) for each given example. With BMO, fewer queries are required on average for a label to meet the threshold c compared to random selection of oracles. Thus, for a given query budget, BMO is expected to obtain more examples with labels of high quality due to the selection of best oracles. As a result, more accurate models can be built.

We conduct extensive experiments on the UCI datasets by generating various types of oracles. The results show that our new algorithm BMO is robust, and performs well with the different types of oracles. The reason is that BMO can guarantee the label quality by querying oracles repeatedly and ensure the best oracles can be queried. As far as we know, this is the first work that proposes this new active learning paradigm.

The rest of this paper is organized as follows. We review related works in Section 2. Section 3 introduces the learning paradigm and the calculation of certainty after querying multiple oracles. We present our learning algorithm, BMO, in Section 4 and the experiment results in Section 5. We conclude our work in Section 6.

2 Previous Works

Labeling each example with multiple oracles has been studied when labeling is not perfect in supervised learning [5,6,7]. Some principled probabilistic solutions have been proposed on how to learn and evaluate the multiple-oracle problem.

However, as far as we know, few of them can guarantee the labeling quality to be equal to or greater than a given threshold c, which can be guaranteed in our work.

Other recent works related to multiple oracles have some assumptions which may be too strong and unrealistic. One assumption is that the noise of oracles is equally distributed [3]. The other type of assumption is that the noise level of different oracles are different as long as they do not change over time [4,8]. Their works estimate the noise level of different oracles during the learning process and prefer querying the oracles with low noise levels. However, it is ubiquitous that the quality of an oracle is example-dependent. In this paper, we remove all the assumptions and allow the noise level of oracles vary among different examples.

Active learning on the data with example-dependent noise level was studied in [9]. However, it focuses on how to choose examples considering the tradeoff between more informative examples and examples with lower noise level.

3 c-Certainty Labeling

C-Certainty labeling is based on the assumption that oracles can return both labels and their confidences in the labelings. For this study, we define *confidence* formally first here. *Confidence* for labeling an example x is the probability that the label given by an oracle is the same as the true label of x. We assume that the confidences of oracles on any example are greater than 0.5[1].

By using the labels and confidences given by oracles, we guarantee that the label certainty of each example can meet the threshold c ($c \in (0.5, 1]$) by querying oracles repeatedly (called *c-certainty labeling*). That is, a label is valid if its certainty is or equal to than c. Otherwise, more queries would be issued to different oracles to improve the certainty.

How to update the label certainty of an example x after obtaining a new answer from an oracle? Let the set of previous $n-1$ answers be A^{n-1}, and the new answer be A_n in the form of (P, f_n), where P indicates positive and f_n is the confidence. The label certainty of x, $C(T_P|A^n)$, can be updated with Formula 1 (See Appendix for the details of its derivation).

$$C(T_P|A^n) = \begin{cases} \frac{p(T_P)\times f_n}{p(T_P)\times f_n + p(T_N)\times(1-f_n)}, & \text{if } n = 1 \text{ and } A_n = \{P, f_n\} \\ \\ \frac{C(T_P|A^{n-1})\times f_n}{C(T_P|A^{n-1})\times f_n + (1-C(T_P|A^{n-1}))\times(1-f_n)}, & \text{if } n > 1 \text{ and } A_n = \{P, f_n\}, \end{cases} \tag{1}$$

where T_P and T_N are the true positive and negative labels respectively. Formula 1 can be applied directly when A_n is positive (i.e., $A_n = \{P, f_n\}$); while for a negative answer, we can transform it as $A_n = \{N, f_n\} = \{P, (1-f_n)\}$ such that

[1] This assumption is reasonable, as usually oracles can label examples more correctly than random.

Formula 1 is also applicable. In addition, Formula 1 is for calculating the certainty of x to be positive. If $C(T_P|A^n) > 0.5$, the label of x is positive; otherwise, the label is negative and the certainty is $1 - C(T_P|A^n)$. With Formula 1, the process of querying oracles can be repeated for x until $max(C(T_P|A^n), 1 - C(T_P|A^n))$ is greater than or equal to c.

However, from Formula 1 we can see that the certainty, $C(T_P|A^n)$, is not monotonic. It is possible that the certainty dangles around and is always lower than c. For example, in paper reviewing, if the labels given by reviewers are with low confidence or alternating between positive and negative, the certainty may not be able to reach the threshold c even many reviewers are requested.

To guarantee that the threshold c is reachable, we will propose an effective algorithm to improve the efficiency of selecting oracles.

4 BMO (Best-Multiple-Oracle) with c-Certainty

To improve the querying efficiency, the key issue is to select the best oracle for *every* given example. This is very different from the case when the noise level is example-independent [4,8], as in our case the performance of each oracle varies on labeling different examples.

4.1 Selecting the Best Oracle

How to select the best oracle given that the noise levels are example-dependent? The basic idea is that an oracle can probably label an example x with high confidence if it has labeled x_j confidently and x_j is close to x. This idea is reasonable as the confidence distribution (expertise level) of oracles is usually continuous, and does not change abruptly. More specifically, we assume that each of the m oracle candidates $(O_1, \cdots, O_m)$ has labeled a set of examples $\mathcal{E}_i$ $(1 \leq i \leq m)$. E_{k_i} $(1 \leq i \leq m)$ is the set of k ($k = 3$ in our experiment) nearest neighbors of x in $\mathcal{E}_i$ $(1 \leq i \leq m)$. BMO chooses the oracle O_i such that examples in E_{k_i} are of high confidence and close to the example x. The potential confidence for each oracle in labeling x can be calculated with Formula 2.

$$P_{c_i} = \frac{\frac{1}{k} \times \sum_{j=1}^{k} f_{x_j}^{o_i}}{1 + \frac{1}{k} \times \sum_{j=1}^{k} |x - x_j|}, \tag{2}$$

where $x_j \in E_{k_i}$, $f_{x_j}^{o_i}$ is the confidence of oracle O_i in labeling x_j, and $|x - x_j|$ is the Euclidean distance between x_j and x. The numerator of Formula 2 is the average confidence of the k nearest neighbors of x. The last item in the denominator is the average distance, and the 1 is added to prevent the denominator from being zero. High confidence and short distance indicate that the oracle O_i will more likely label x with a higher confidence. Thus, BMO selects the oracle O_i if $i = \arg\max_i(P_{c_1}, \cdots, P_{c_i}, \cdots, P_{c_m})$.

4.2 Active Learning Process of BMO

BMO is a wrapper learning algorithm, and it treats the strategy of selecting examples to label as a black box. Any existing query strategies in active learning, such as uncertainty sampling [10], expected error reduction [11] and the density-weighted [12] method can be fit in easily.

We assume that BMO starts with an empty training set, and the learning process is as follows. For an example x_i ($x_i \in \mathcal{E}_u$) selected by an example-selecting strategy (e.g.,uncertain sampling), BMO selects the best oracle among the ones that have not been queried for x_i yet to query, and updates the label certainty of x_i with Formula 1. This process repeats until the certainty meets the threshold c. Then BMO adds x_i into its labeled example set $\mathcal{E}_l$. This example-labeling process continues until certain stop criterion is met (such as the predefined query budget is used up in our experiment). (See Algorithm 1 for details.)

Algorithm 1. BMO (Best Multiple Oracles)

```
Input: Unlabeled data: $\mathcal{E}_u$; oracles: $\mathcal{O}$; oracles queried: $\mathcal{O}_q$; threshold: $c$;
       queries budge: $budget$;
Output: labeled example set $\mathcal{E}_l$
begin
    while $budget > 0$ do
        $x_i \leftarrow$ selection with uncertain sampling ($x_i \in \mathcal{E}_u$);
        $\mathcal{O}_q \leftarrow null$;   $certainty \leftarrow 0$;
        while $certainty < c$ do
            for $each\ O_i \in (\mathcal{O} - \mathcal{O}_q)$ do
                $Po\_c_i \leftarrow$Formula 2;
            end
            $O_m \leftarrow$ oracle with maximal $P_c$;
            $certainty \leftarrow$ update with Formula 1;
            $\mathcal{O}_q \leftarrow \mathcal{O}_q \cup O_m$; $budget \leftarrow budget - 1$;
        end
        $\mathcal{E}_l \leftarrow \mathcal{E}_l \cup anditscertainty$;
        update the current model;
    end
    return $\mathcal{E}_l$;
end
```

By selecting the best oracles, BMO can improve the label certainty of a given example to meet the threshold c with only a few queries (See Section 5). That is, more labeled examples can be obtained for a predefined query budget compared to random selection of oracles. Thus, the model built is expected to have better performance.

5 Experiments

In our experiment, to compare with BMO, we implement two other learning strategies. One is *Random selection of Multiple Oracles* (RMO). Rather than selecting the best oracle in BMO, RMO selects oracles randomly to query for a given example and repeats until the label certainty is greater than or equal to c. The other strategy is *Random selection of Single Oracle* (RSO). RSO queries for each example only once without considering c, which is similar to traditional active learning algorithms.

Since RSO only queries one oracle for each example, it will have the most labeled examples for a predefined query budget but with the highest noise level. To reduce the negative effect of noisy labels, we *weight* all labeled examples according to their label certainty when building final models. To make all the three strategies comparable, we also use weighting in BMO and RMO. In addition, all the three algorithms take uncertain sampling as the example-selecting strategy and decision tree (J48 in WEKA [13]) as their base learners. The implementation is based on the WEKA source code.

The experiment is conducted on UCI datasets [14], including abolone, anneal, cmc_new, credit, mushroom, spambase and splice, which are commonly used in the supervised learning research. As the number of oracles cannot be infinite in real world, we only generate 10 oracles for each dataset. If an example has been presented to all the 10 oracles, the label and the certainty obtained will be taken in directly. The threshold c is predefined to be 0.8 and 0.9 respectively. The experiment results presented are the average of 10 runs, and t-test results are of 95% confidence.

In our previous discussion, we take the confidence given by an oracle as the true confidence. However, in real life, oracles may overestimate or underestimate themselves intentionally or unintentionally. If the confidence given by an oracle O does not equal the true confidence, we call *O an unfaithful oracle*; otherwise, it is faithful. To observe the robustness of our algorithm, we conduct our empirical studies with both faithful and unfaithful oracles[2] in the following.

5.1 Results on Faithful Oracles

As no oracle is provided for the UCI data, we generate a faithful oracle as follows. Firstly, we select one example x randomly as an "expertise center" and label it with the highest confidence. Then, to make the oracle faithful, we calculate the Euclidean distance from each of the rest examples to x, and assign them confidences based on the distances. The further the distance is, the lower confidence the oracle has in labeling the example. Noise is added into labels according to the confidence level. Thus the oracle is faithful.

The confidence is supposed to follow a certain distribution. We choose three common distributions, linear, normal and dual normal distributions. Linear distribution assumes the confidence reduces linearly as the distance increases. For normal distribution, the reduction of confidence follows the probability density function $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{x^2}{2\sigma^2}) - 0.55$. Dual normal distribution indicates that

[2] Actually it is difficult to model the behaviors of unfaithful oracles with a large confidence deviation. In our experiment, we show that our algorithm works well given unfaithful oracles slightly deviating from the true confidence.

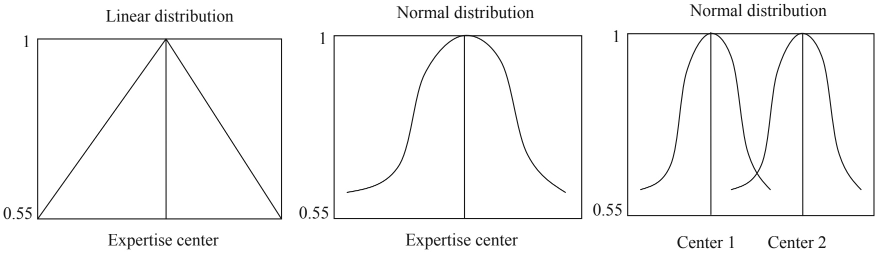


Fig. 1. Three distributions

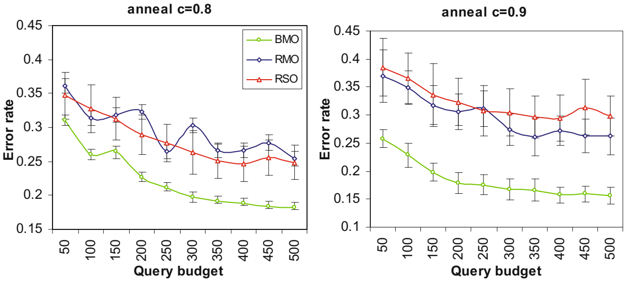


Fig. 2. Error rate on faithful oracles

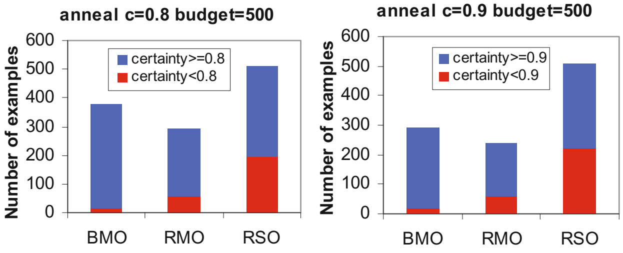


Fig. 3. The number of examples and label quality

the oracle has two "expertise centers" (see Figure 1). As mentioned earlier, we generate 10 oracles for each dataset in this experiment. Among the 10 oracles, three of them follow the linear distribution, three the normal distribution and four the dual normal distribution.

Due to the similar results of different datasets, we only show the details of one dataset (anneal) in Figure 2 and a summary of the comparison afterwards. Figure 2 shows the testing error rates of BMO, RMO and RSO for the threshold 0.8 (left) and 0.9 (right) respectively. The x axis indicates the query budgets while the y axis represents the error rate on test data. On one hand, as we expected that, for both thresholds 0.8 and 0.9, the error rate of BMO is much lower than that of RMO and RSO for all different budgets, and the performances of the latter two are similar. On the other hand, the curve of RMO when $c = 0.8$ is not as smooth as the other ones.

The different performances of the three learning strategies can be explained by two factors, the noise level and the number of examples. Due to limited space, we only show how the two factors affect the performances through one dataset (anneal) when the query budget is 500 in Figure 3.

Figure 3 shows that on average BMO only queries about 1.4 ($c = 0.8$) and 1.7 ($c = 0.9$) oracles for each example; while RMO queries more oracles (1.7 and 2.0). That is, BMO obtains more labeled examples than RMO for a given budget. Moreover, the examples labeled by BMO have much higher label certainty than that by RMO[3]. On the other hand, the examples labeled by RSO is much more noisy than BMO (i.e., the red portion is much larger). It is the noise that deteriorates the performance of RSO. Thus, BMO outperforms the other two strategies because of its guaranteed label quality and the selection of the best oracles to query.

By looking closely into the curves in Figure 2, we find that the curve of RMO when $c = 0.8$ is not as smooth as the other ones. The reason is that RMO of $c = 0.8$ has fewer labeled examples when compared to BMO and RSO of $c = 0.8$ and has more noise when compared to that of $c = 0.9$. Fewer examples make the model learnt more sensitive to the quality of each label; while the label quality of $c = 0.8$ is not high enough. Thus, the stability of RMO when $c = 0.8$ is weakened.

In addition, we also show the t-test results in terms of the error rate on all the seven UCI datasets in Table 1. As for each dataset 10 different query budgets are considered, the total times of t-test for each group is 70. Table 1 shows that BMO wins RMO 94 times out of 140 ($c = 0.8$ and $c = 0.9$) and wins RSO 86 out of 140 without losing once. It is clear that BMO outperforms RMO and RSO significantly.

Table 1. T-test results on 7 datasets with 10 different budgets

	BMO vs. RMO		BMO vs. RSO		RMO vs. RSO	
	c=0.8	c=0.9	c=0.8	c=0.9	c=0.8	c=0.9
Win	43	51	40	42	0	9
Draw	27	19	30	38	70	51
Lose	0	0	0	0	0	10

In summary, with faithful oracles, the experiment results show that BMO does work better by guaranteeing the label quality and selecting the best oracles to query. On the other hand, even though RMO also can guarantee the label quality, its strategy of randomly selecting oracles reduces the learning performance. Furthermore, the results of RSO illustrate that weighting with the label quality may reduce the negative influence of noise but still its effect is limited.

[3] Some of the examples still have certainty lower than c due to the limited oracles in our experiment.

5.2 Results on Unfaithful Oracles

Unfaithful oracles are generated for each dataset by building models over 20% of the examples. More specifically, to generate an oracle, we randomly select one example x as an "expertise center", and sample examples around it. The closer an example x_i is to x, the higher the probability it will be sampled with. Thus, the oracle built on the sampled examples can label the examples closer to x with higher confidences. The sampling probability follows exactly the same distribution in Figure 1. For each data set, 10 oracles are generated and three follow the linear distribution, three the normal distribution and four the dual normal distribution.

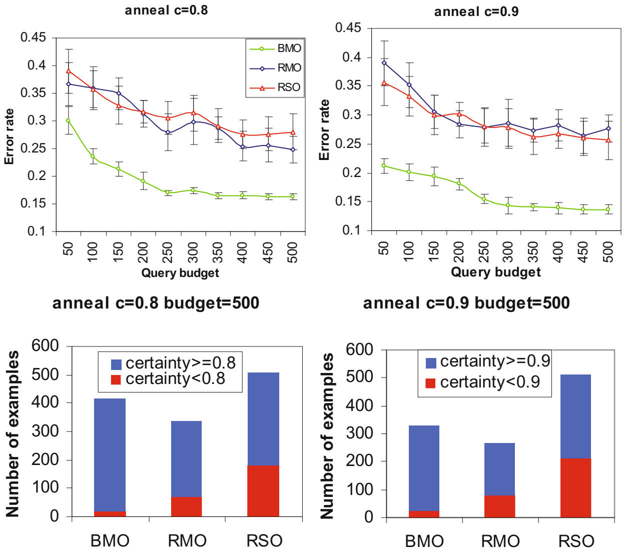


Fig. 4. Experiment results on unfaithful oracles

As sampling rate declines with the increasing distance, the oracle built may fail to give true confidence for the examples that are far from the "center". As a result, the oracle is unfaithful. That is, the oracles are unfaithful due to "insufficient knowledge" rather than "lying" deliberately.

We run BMO, RMO and RSO on the seven UCI datasets and show the testing error rates and the number of labeled examples on one data set (anneal) in Figure 4 and a summary on all the datasets afterwards. It is surprising that the performances of BMO on unfaithful oracles are similar to that on faithful oracles. That is, the error rate of BMO is much less than that of RMO and RSO, and the latter two are similar. The examples labeled by BMO are more than that by RMO and its label quality is higher than that of both RMO and SMO, which are also similar to that on faithful oracles.

The comparison shows clearly that BMO is robust even for unfaithful oracles. The reason is that BMO selects the best multiple oracles to query, and it is unlikely that all the best oracles are unfaithful at the same time as our unfaithful oracles do not "lie" deliberately as mentioned. Thus, BMO still performs well.

Table 2 shows the t-test results on 10 different query budgets for all the seven UCI datasets. We can see that BMO wins RMO 95 times out of 140 and wins RSO 98 out of 140, which indicates that BMO works significantly better than RMO and RSO under most of the circumstances. However, BMO loses to RMO 19 times and RSO 10 times, which are different from the results on faithful oracles. Thus, even though BMO is robust, still it works slightly worse on unfaithful oracles than on faithful ones.

Table 2. T Test results for all datasets and budgets on unfaithful oracles

	BMO vs. RMO		BMO vs. RSO		RMO vs. RSO	
	c=0.8	c=0.9	c=0.8	c=0.9	c=0.8	c=0.9
Win	53	42	53	45	12	0
Draw	6	22	17	15	50	50
Lose	11	8	0	10	8	20

In summary, BMO is robust for working with unfaithful oracles, even though its good performance may be reduced slightly. This property is crucial for BMO to be applied successfully in real applications.

6 Conclusion

In this paper, we proposed a novel active learning paradigm, c-certainty learning, in which oracles can return both labels and confidence. Under this new paradigm, the label quality is guaranteed to be greater than or equal to a given threshold c by querying multiple oracles. Furthermore, we designed the learning algorithm BMO to select the best oracles to query so that the threshold c can be met with fewer queries compared to selecting oracles randomly. Empirical studies are conducted for both faithful and unfaithful oracles. The results show that BMO works robustly and outperforms other active learning strategies significantly on both faithful and unfaithful oracles, even though its performance can be affected slightly by unfaithful oracles.

References

1. Balcan, M., Beygelzimer, A., Langford, J.: Agnostic active learning. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 65–72. ACM (2006)
2. Settles, B.: Active Learning Literature Survey. Machine Learning 15(2), 201–221 (1994)

3. Sheng, V., Provost, F., Ipeirotis, P.: Get another label? improving data quality and data mining using multiple, noisy labelers. In: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 614–622. ACM (2008)
4. Donmez, P., Carbonell, J., Schneider, J.: Efficiently learning the accuracy of labeling sources for selective sampling. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 259–268. ACM (2009)
5. Raykar, V., Yu, S., Zhao, L., Jerebko, A., Florin, C., Valadez, G., Bogoni, L., Moy, L.: Supervised Learning from Multiple Experts: Whom to trust when everyone lies a bit. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 889–896. ACM (2009)
6. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.: Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 254–263. Association for Computational Linguistics (2008)
7. Sorokin, A., Forsyth, D.: Utility data annotation with amazon mechanical turk. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2008, pp. 1–8. IEEE (2008)
8. Zheng, Y., Scott, S., Deng, K.: Active learning from multiple noisy labelers with varied costs. In: 2010 IEEE International Conference on Data Mining, pp. 639–648. IEEE (2010)
9. Du, J., Ling, C.: Active learning with human-like noisy oracle. In: 2010 IEEE International Conference on Data Mining, pp. 797–802. IEEE (2010)
10. Lewis, D., Gale, W.: A sequential algorithm for training text classifiers. In: Proceedings of the 17th Annual International ACM SIGIR Conference, pp. 3–12. Springer-Verlag New York, Inc. (1994)
11. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: Machine Learning-International Workshop then Conference, pp. 441–448. Citeseer (2001)
12. Settles, B., Craven, M., Ray, S.: Multiple-instance active learning. In: Advances in Neural Information Processing Systems (NIPS). Citeseer (2008)
13. WEKA Machine Learning Project, "Weka", http://www.cs.waikato.ac.nz/~ml/weka
14. Asuncion, A., Newman, D.: UCI machine learning repository (2007), http://www.ics.uci.edu/~mlearn/mlrepository.html

Appendix: Derivation of Formula 1

$$\begin{aligned}
&C(T_P|A^n) \\
&= \frac{P(A^n|T_P) \times P(T_P)}{P(A^n)} \\
&= \frac{P(A^{n-1}, A_n|T_P) \times P(T_P)}{P(A^n)} \\
&= \frac{P(A^{n-1}|T_P) \times P(T_P) \times P(A_n|T_P) \times P(A^{n-1})}{P(A^{n-1}) \times P(A^n)} \\
&= C(T_P|A^{n-1}) \times p(A_n|T_P) \times \frac{p(A^{n-1})}{p(A^n)}
\end{aligned} \tag{3}$$

The last item in Equation 3 can be further transformed as follows.

$$\begin{aligned}
&\frac{p(A^{n-1})}{p(A^n)} \\
&= \frac{p(A^{n-1})}{p(A^n|T_P) \times p(T_P) + p(A^n|T_N) \times p(T_N)} \\
&= \frac{p(A^{n-1})}{p(A^{n-1}|T_P) \times p(T_P) \times p(A_n|T_P) + p(A^{n-1}|T_N) \times p(T_N) \times p(A_n|T_N)} \\
&= \frac{1}{(C(T_P|A^{n-1})) \times p(A_n|T_P) + C(T_N|A^{n-1}) \times p(A_n|T_N))}
\end{aligned}$$

As $A_n = (P, f_n)$,

$$\begin{aligned}
&C(T_P|A^n) \\
&= \frac{C(T_P|A^{n-1}) \times p(A_n|T_P)}{C(T_P|A^{n-1}) \times p(A_n|T_P) + (1 - C(T_P|A^{n-1})) \times p(A_n|T_N)} \\
&= \frac{C(T_P|A^{n-1}) \times f_n}{C(T_P|A^{n-1}) \times f_n + (1 - C(T_P|A^{n-1})) \times (1 - f_n)}
\end{aligned}$$

A Term Association Translation Model for Naive Bayes Text Classification

Meng-Sung Wu and Hsin-Min Wang

Institute of Information Science, Academia Sinica, Taipei, Taiwan
{wums,whm}@iis.sinica.edu.tw

Abstract. Text classification (TC) has long been an important research topic in information retrieval (IR) related areas. In the literature, the bag-of-words (BoW) model has been widely used to represent a document in text classification and many other applications. However, BoW, which ignores the relationships between terms, offers a rather poor document representation. Some previous research has shown that incorporating language models into the naive Bayes classifier (NBC) can improve the performance of text classification. Although the widely used N-gram language models (LM) can exploit the relationships between words to some extent, they cannot model the long-distance dependencies of words. In this paper, we study the term association modeling approach within the translation LM framework for TC. The new model is called the term association translation model (TATM). The innovation is to incorporate term associations into the document model. We employ the term translation model to model such associative terms in the documents. The term association translation model can be learned based on either the joint probability (JP) of the associative terms through the Bayes rule or the mutual information (MI) of the associative terms. The results of TC experiments evaluated on the Reuters-21578 and 20newsgroups corpora demonstrate that the new model implemented in both ways outperforms the standard NBC method and the NBC with a unigram LM.

Keywords: Term association, mutual information, Bayes, translation language model, text classification.

1 Introduction

Text classification (TC) is the task of classifying documents into a set of predefined categories. It has long been an important research topic in information retrieval (IR). Many statistical classification methods and machine learning (ML) techniques have been developed to TC, such as the naive Bayes classifier [12], the support vector machines [10], the k-nearest neighbor method [20], and the boosting method [16]. In addition, text classification based on term associations [1] is also a promising approach. The performance of text classification highly depends on the document representation. Most of the existing methods represent a document using a vector space model (VSM) or a language model (LM). Generally, the bag-of-words (BoW) method is a widely used data representation

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 243–253, 2012.

in IR and TC. Under this scheme, each document is modeled as a vector with a dimension equal to the size of the dictionary, and each element of the vector denotes the frequency that a word appears in the document. Basically, all the words are treated independently.

One of the important restrictions in most of the existing TC methods may lie in that the individual terms are usually too general and that these methods do not consider the associations between words in the documents. In some cases of TC, individual words are not sufficient to represent the accurate information of the document. For example, a document with "shuttle launch" may be assumed to belong to the "ball game" class. However, if the word "NASA" is an association term, it is very likely that the document should be assigned to the "aeronautics" class.

It is well-known that the relationships between words are very important for statistical language modeling. Using LM for TC has been studied recently [2,14]. Although N-gram LM can exploit the relationships between words, they only consider the dependencies of neighboring words [5]. For example, the trigram LM is unable to characterize word dependence beyond the span of three successive words. In [22], the trigram LM was improved by integrating with the trigger pairs, which extract the word relationships from the sequence of historical words. Nevertheless, a trigger pair is word order dependent. In other words, a word can only be triggered by the previous context. Recent studies have revealed that modeling term associations could provide richer semantics of documents for LM and IR [4,18,19]. Cao et al. [4] integrated the word co-occurrence information and the WordNet information into language models. Wei and Croft [18] investigated the use of term associations to improve the performance of LM-based IR. In [19], the word associations were integrated into the topic modeling paradigm. Adding word associations to represent a document inevitably increases the model's complexity, but the new information reduces the ambiguity mentioned above. Generally, any set of words co-occur in the contexts can be considered having a strong association and collected as the associative words, e.g., "uneven bars" and "balance" in the class of gymnastics and "aerofoil" and "jet engine" in the class of airplane transportation. However, the associative words are not necessary to co-occur in a document. We believe that a language model considering term associations would be definitely more useful in TC.

In this paper, we propose a novel model for text classification by incorporate the strengths of term associations into the translation LM framework. Different from the traditional TC techniques and algorithms in the literature, we model the associations between words existing in the documents of a class. To discover the associative terms in the documents, we learn the translation language model based on the joint probability (JP) of the associative terms through the Bayes rule and based on the mutual information (MI) of the associative terms.

The remainder of this paper is organized as follows. In Section 2, we briefly review the framework of the naive Bayes classifier and language models. The proposed models for text classification are presented in Section 3. Experimental setup and results are discussed in Section 4. Finally, we give the conclusions in Section 5.

2 Related Work

2.1 Terminology

We begin by defining the notation and terminology in this paper. A **word** or **term** is a linguistic building block for text. A word is denoted by $w \in V = \{1, 2, \ldots, |V|\}$, where $|V|$ is number of distinct words/terms. A **document**, represented by $\mathbf{d} = \{w_1, \cdots, w_{n_d}\}$, is an ordered list of n_d words. A query, denoted by $\mathbf{q} = \{q_1, \cdots, q_T\}$, is a string of T words. A **collection** of documents is denoted by $D = \{\mathbf{d}_1, \cdots, \mathbf{d}_{|D|}\}$, where $|D|$ is the number of documents in collection D. A **background model**, denoted by $\mathcal{M}_B$, is the language model estimated in collection D. A set of **class labels** is denoted by $C = \{c_1, \cdots, c_{|C|}\}$, where $|C|$ is the number of distinct classes. A **LM** $\mathcal{M}$ is a probability function defined on a set of word strings. This includes the important special case of the probability $P(w|\mathcal{M})$ of a word w. A **class LM**, denoted by $\mathcal{M}_C$, is the language model estimated based on class c.

2.2 Naive Bayes Classifier

The naive Bayes classifier (NBC) is a popular machine learning technique for text classification. The method assumes a probabilistic generative model for text. A common and simple representation of a document in TC is the bag of words (BoW) model. The model ignores the word order and just captures the number of occurrences of each word in the document. The NBC classifies a document through two stages: the learning stage and the classifying stage. It is assumed that the probability of each word in a document is independent to that of other words, and each document is drawn from a multinomial distribution of words. In the learning stage, the naive Bayes classifier estimates the conditional probability $P(c|d)$, which represents the probability that a document c belongs to a class d. Using the Bayes rule, we have

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} = \frac{P(d|c)P(c)}{\sum_c P(d|c)P(c)}, \tag{1}$$

where $P(d|c)$ is the likelihood of document d under class c. By assuming that all words in d are independent of each other, $P(d|c)$ can be further decomposed into the product of individual feature (word) probabilities as follows

$$P(d|c) = \prod_{w \in V} P(w|c). \tag{2}$$

The word probability $P(w|c)$ and the class prior probability $P(c)$ are estimated from the training documents with Laplace smoothing as follows

$$P(w|c) = \frac{1 + n(w,c)}{|V| + N(c)}, \tag{3}$$

$$P(c) = \frac{1 + n(d,c)}{|C| + |D|}, \tag{4}$$

where $n(w, c)$ is the number of times word w occurs in the training documents that belong to class c; $N(c)$ is the total number of words in the training documents that belong to class c; $n(d, c)$ denotes the number of documents that belong to class c; and $|D|$ is total number of training documents.

Several extensions of the naive Bayes classifier have been proposed. For example, Nigam et al. [13] combined the Expectation-Maximization (EM) algorithm and the naive Bayes classifier to learn from both labeled and unlabeled documents in a semi-supervised manner. More recently, Dai et al. [7] proposed a transfer learning algorithm to learn the naive Bayes classifier for text classification, which allowed the distributions of the training and test data to be different. However, these methods all assume that the words in a document are independent of each other; hence, they cannot cope well with the term dependence and association.

2.3 Language Models for Information Retrieval

Statistical language modeling plays an important role in automatic speech recognition (ASR) and IR. Most ASR systems are built by combining the N-gram language model and the acoustic hidden Markov model (HMM) to predict the best word sequence corresponding to an input speech utterance. In an IR system, the word sequence of an input query is adopted to retrieve the relevant text documents. In Ponte and Croft's work that applied LM in IR [15], the retrieval performance was improved by statistical modeling of natural language. According to the maximum a posteriori decision rule, the ranking function $f(\cdot)$ is established as a posterior probability,

$$\hat{\mathbf{d}} = \arg\max_{\mathbf{d}_m} f(\mathbf{q}, \mathbf{d}_m) = \arg\max_{\mathbf{d}_m} P(\mathbf{d}_m|\mathbf{q}) = \arg\max_{\mathbf{d}_m} P(\mathbf{q}|\mathbf{d}_m)P(\mathbf{d}_m). \quad (5)$$

Assuming that the documents $\{\mathbf{d}_1, \cdots, \mathbf{d}_{|D|}\}$ have an equal prior probability of relevance, the ranking can be done according to the likelihood of the N-gram language model

$$P(\mathbf{q}|\mathbf{d}_m) = P(q_1, \cdots, q_T|\mathbf{d}_m) = \prod_{t=1}^{T} P(q_t|q_{t-n+1}^{t-1}, \mathbf{d}_m), \quad (6)$$

where each word q_t only depends on its $n-1$ historical words $q_{t-n+1}^{t-1} = \{q_{t-n+1}, \cdots, q_{t-1}\}$. $P(q_t|q_{t-n+1}^{t-1}, \mathbf{d}_m)$ can be estimated according to the maximum likelihood (ML) criterion as follows,

$$P_{\mathrm{ML}}(q_t|q_{t-n+1}^{t-1}, \mathbf{d}_m) = \frac{c(q_{t-n+1}^{t}, \mathbf{d}_m)}{c(q_{t-n+1}^{t-1}, \mathbf{d}_m)}, \quad (7)$$

where $c(q_{t-n+1}^{t}, \mathbf{d}_m)$ denotes the number of times that word q_t follows the historical words q_{t-n+1}^{t-1} in document $\mathbf{d}_m$ and $c(q_{t-n+1}^{t-1}, \mathbf{d}_m)$ denotes the number of times that the historical words q_{t-n+1}^{t-1} occur in document $\mathbf{d}_m$. The unigram document model is generally adopted in the IR community [15]. However, the document terms are often too few to train a reliable ML-based model because the unseen words lead to zero unigram probabilities. Zhai and Lafferty [21] have used several smoothing methods to deal with the data sparseness problem in LM-based IR.

Since previous research [4,18,19] have shown that some relationships exist between words, we utilize them in the document model rather than using the traditional unigram document model for text classification.

3 The Term Association Translation Models

3.1 Language Models for Text Classification

LM was first introduced to TC by Peng and Schuurmans [14]. The score of a class c for a given document d can be estimated by (1). Then, the class of the document can be decided as follows

$$c^* = \arg\max_{c \in C} P(d|c) = \arg\max_{c \in C} P(d|c)P(c). \tag{8}$$

Assuming that $P(c)$ is uniformly distributed and applying the unigram class LM in the task, the decision can be rewritten as

$$\begin{aligned} c^* &= \arg\max_{c \in C} P(d|c) \\ &= \arg\max_{c \in C} \prod_{i=1}^{n_d} P(w_i|c). \end{aligned} \tag{9}$$

The traditional naive Bayes classifier usually uses Laplace smoothing to deal with the zero probability problem. However, some previous research has shown that it is not as effective as the smoothing methods for language modeling [2,14]. Therefore, we can interpolate a unigram class LM with the unigram collection background model by using the *Jelinek-Mercer* smoothing method as follows,

$$P(w_i|\mathcal{M}_C) = \lambda P(w_i|c) + (1-\lambda)P(w_i|\mathcal{M}_B), \tag{10}$$

where λ can be tuned empirically. In this paper, the method based on (10) is denoted as NBC-UN, and λ is set to 0.5.

In order to discover the association between two terms w_i and w, we are interested in $P_t(w_i|w)$, the probability that word w_i will occur given that w occurs. The term translation probability $P_t(w_i|w)$ is different from the bigram probability $P(w_i|w)$ in that the words w_i and w are not limited to occur in order and adjacently in the former. Then, the term association information can be integrated into the unigram class model as follows,

$$P(w_i|c) = \sum_{w \in c} P(w_i|w)P(w|c), \tag{11}$$

where $P(w|c)$ reflects the distribution of words in the training documents of class c, which can be computed via the maximum likelihood estimate. By replacing $P(w_i|c)$ in (10) with the one computed by (11), we have

$$P(w_i|\mathcal{M}_C) = \lambda[\sum_{w \in c} P(w_i|w)P(w|c)] + (1-\lambda)P(w_i|\mathcal{M}_B). \quad (12)$$

The model in (12) is obviously more computationally intensive than the model in (9). Therefore, we need to build a global term translation model for all classes and the word probability distribution for each class beforehand. To discover the associative terms in the training documents, we learn the translation LM based on the joint probability of the associative terms through the Bayes rule and based on the mutual information (MI) of the associative terms.

3.2 Translation Model Estimation Using Joint Probability Model

This section describes our first way of constructing the term translation probability $P_t(w_i|w)$. By definition, we can express the conditional probability as the joint probability of words w_i and w over the probability of word w

$$P_t(w_i|w) = \frac{P(w_i, w)}{P(w)}, \quad (13)$$

where the join probability of w_i and w can be expressed as

$$P(w_i, w) = \sum_c P(w_i, w|c)P(c) = \sum_c P(w_i|c)P(w|c)P(c), \quad (14)$$

if w_i and w are assumed sampled independently and identically from the unigram class model c, and the probability of w can be expressed as

$$P(w) = \sum_c P(w|c)P(c). \quad (15)$$

After re-normalizing $P(w_i, w)$ in (14) and $P(w)$ in (15), and considering a uniform prior $P(c)$, we obtain

$$P_t(w_i|w) = \frac{\sum_c P(w_i|c)P(w|c)}{\sum_c P(w|c)}. \quad (16)$$

The method based on (12) with $P_t(w_i|w)$ computed by (16) is denoted as TATM-JP (the *term association translation model* estimated by the *joint probability* of terms).

3.3 Translation Model Estimation Based on Mutual Information

Our second way of constructing the term translation probability $P_t(w_i|w)$ is based on the mutual information (MI). In information theory, the MI of two random variables is a quantity that measures their mutual dependence. MI is a good measure to assess how two words are related to each other [6,22]. We use the average mutual information (AMI) [22] to measure the strength of the association between words w_i and w. The AMI between w_i and w is defined as follows

$$\begin{aligned} AMI(w_i, w) = {} & P(w_i, w) log \frac{P(w_i, w)}{P(w_i)P(w)} + P(w_i, \bar{w}) log \frac{P(w_i, \bar{w})}{P(w_i)P(\bar{w})} \\ & + P(\bar{w}_i, w) log \frac{P(\bar{w}_i, w)}{P(\bar{w}_i)P(w)} + P(\bar{w}_i, \bar{w}) log \frac{P(\bar{w}_i, \bar{w})}{P(\bar{w}_i)P(\bar{w})} \end{aligned} \quad (17)$$

where $P(w_i, w)$ is estimated as the ratio of the number of documents that contain both w_i and w, i.e., $c_d(w_i, w)$, and the total number of documents $|D|$ as follows

$$P(w_i, w) = \frac{c_d(w_i, w)}{|D|}; \quad (18)$$

$P(w_i, \bar{w})$ is computed by

$$P(w_i, \bar{w}) = \frac{c_d(w_i) - c_d(w_i, w)}{|D|}, \quad (19)$$

where $c_d(w_i)$ is the number of documents that contain w_i; $P(w)$ is estimated as the ratio of the number of documents that contain w and the total number of documents; $P(\bar{w})$ is estimated as the ratio of the number of documents that do not contain w and the total number of documents; and the other probabilities are estimated in a similar way. According to [11], the term translation probability $P_t(w_i|w)$ can be calculated by normalizing the mutual information score as follows

$$P_t(w_i|w) = \frac{AMI(w_i, w)}{\sum_{w_j} AMI(w_j, w)}. \quad (20)$$

If the two words w_i and w tend to associate with each other, the probability would be higher. The method based on (12) with $P_t(w_i|w)$ computed by (20) is denoted as TATM-MI (the *term association translation model* estimated based on the *mutual information* of terms).

4 Experiments

4.1 Corpora

We evaluate the proposed TC methods on two standard document collections: Reuters-21578 (Reuters)[1] and 20 Newsgroups (20NG)[2]. According to the ModApte split, the Reuters corpus is separated into 7,194 documents for training and 2,788 documents for testing. 135 categories have been defined, but only 118

[1] http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

[2] http://people.csail.mit.edu/jrennie/20Newsgroups/

Table 1. Statistics of the Reuters collection

Category	Training Set	Test Set
earn	2877	1087
acq	1650	719
money-fx	538	179
grain	433	149
crude	389	189
trade	369	118
interest	347	131
wheat	212	71
ship	197	89
corn	182	56

categories have documents assigned to them. Following Debole and Sebastiani's work in [8], we consider the most frequent ten categories in the experiments. The 10 categories and the numbers of documents used for training and testing in each category are listed in Table 1. The 20NG dataset is a collection of 19,974 documents collected from 20 different newsgroups. We consider the 20 newsgroups as the 20 categories. For each category, we randomly select 60% of the documents for training and the remaining 40% for testing. Since the 20NG collection distributes roughly evenly across 20 newsgroups, each category has almost the same number of training (or testing) documents.

4.2 Performance Measure

In the following experiments, the performance of text classification is evaluated in terms of the recall (R), precision (P), and F-measure (F), calculated as follows:

$$\text{recall} = \frac{\#\text{ of correct postive predictions}}{\#\text{ of postive examples}}, \tag{21}$$

$$\text{precision} = \frac{\#\text{ of correct postive predictions}}{\#\text{ of postive predictions}}, \tag{22}$$

$$F = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}. \tag{23}$$

To evaluate the average performance across classes, we use the micro-averaged score and macro-averaged score [20]. The micro-averaged score is calculated by mixing together the documents across all the classes. The macro-averaged score is obtained by taking the average of the recall, precision, and F-measure values for each category

4.3 Experimental Results

We compare our term association translation models (TATM-JP and TATM-MI) with the naive Bayes classifier with Laplace smoothing (NBC) and the naive Bayes classifier with the unigram language model (NBC-UN).

Table 2. Experimental results (in F-measure) for the Reuters collection

	NBC	NBC-UN	TATM-JP	TATM-MI
earn	0.814	0.825	0.819	0.824
acq	0.801	0.811	0.801	0.802
money-fx	0.511	0.521	**0.543**	0.539
grain	0.578	0.583	0.616	**0.634**
crude	0.577	0.596	0.610	**0.618**
trade	0.439	0.434	**0.477**	0.465
interest	0.483	0.477	**0.516**	0.502
wheat	0.490	0.506	0.577	**0.603**
ship	0.571	0.583	0.624	**0.639**
corn	0.466	0.468	**0.527**	0.491
micro-averaged	0.709	0.720	0.727	**0.731**

Table 3. The micro/macro-averaged precision, recall, and F-measure of different methods evaluated on the 20NG dataset

	Micro-averaged			Macro-averaged		
	P	R	F	P	R	F
NBC	0.802	0.800	0.801	0.817	0.795	0.806
NBC-UN	0.809	0.807	0.808	0.822	0.802	0.812
TATM-JP	0.818	0.815	0.817	0.827	0.810	0.818
TATM-MI	0.821	0.819	0.820	0.829	0.814	0.821

Table 2 shows the results of text classification experiments evaluated on the Reuters collection. The measure used is the F-measure on the ten most populated Reuters-21578 categories and the micro-averaged F-measure (micro-F) over all categories. Comparing the results of NBC and NBC-UN, it is obvious that using language models improves the classification effectiveness of the naive Bayes classifier. Both proposed methods consistently outperform NBC and respectively perform better than NBC-UN in four out of ten categories. The micro-average F-measure of TATM-MI is 0.731, which is better than that of TATM-JP (0.727), NBC-UN (0.720) and NBC (0.709). The relative improvement in the micro-F by TATM-MI is 3.1% over NBC and 1.5% over NBC-UN.

Table 3 shows the experimental results for the 20NG dataset in terms of the micro/macro-averaged precision, recall, and F-measure. The micro-F of TATM-JP is 0.817 and TATM-MI is 0.82, which is better than that obtained by NBC (0.801) and NBC-UN (0.808). The relative improvement by TATM-JP over NBC and NBC-UN is 2%, and 1.11%, respectively. Similarly, the relative improvement in micro-F by TATM-MI over NBC and NBC-UN is 2.37%, and 1.49%, respectively. The improvements of TATM-JP and TATM-MI over NBC and NBC-UN are statistically significant according to the t-test. In addition, the term association translation model estimating based on the mutual information for all data sets is more efficient than learning the term association translation model by the joint probability. As expected, the performance in micro-F on the 20NG dataset is very similar to that in macro-F because each class has a similar number of training and testing documents. Again, we can see that TATM-MI performs the best.

Several observations can be drawn from the results. First, the performance of text classification can be improved by incorporating language models into the naive Bayes classifier. Second, the proposed document model with term association modeling leads to improvements over NBC and NBC-UN. The new model could be applied to other topic document models.

5 Conclusion and Future Work

The use of term associations for TC has attracted great interest. This paper has presented a new term association translation model, which models term associations, for TC. The proposed model can be learned based on the joint probability of the associative terms through the Bayes rule or based on the mutual information of the associative terms. The experimental results show that the new model learned in either way outperforms the traditional TC methods. For future work, we plan to investigate the effect of the feature selection method [17] for the selection of associative terms. In addition, we will integrate our model into the topic models such as probability latent semantic analysis (PLSA) [9] or latent Dirichlet allocation (LDA) [3] for text classification. Another interesting direction is to combine the term association document model with the relevance-based document model, and apply the combined model in TC.

References

1. Antonie, M.L., Zaiane, O.R.: Text Document Categorization by Term Association. In: Proceedings of IEEE 2002 International Conference on Data Mining (ICDM), pp. 19–26 (2002)
2. Bai, J., Nie, J.Y.: Using language models for text classification. In: Proceedings of the Asia Information Retrieval Symposium, AIRS (2004)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3(4-5), 993–1022 (2003)
4. Cao, G., Nie, J.Y., Bai, J.: Integrating word relationships into language models. In: Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval, pp. 298–305 (2005)
5. Chien, J.T., Wu, M.S., Peng, H.J.: Latent semantic language modeling and smoothing. International Journal of Computational Linguistics and Chinese Language Processing 9(2), 29–44 (2004)
6. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Computational Linguistics 16(1), 22–29 (1990)
7. Dai, W., Xue, G.R., Yang, Q., Yu, Y.: Transferring Naive Bayes Classifiers for Text Classification. In: Proceedings of AAAI Conference on Artificial Intelligence, pp. 540–545 (2007)

8. Debole, F., Sebastiani, F.: An analysis of the relative difficulty of Reuters-21578 subsets. Journal of the American Society for Information Science and Technology 56(2), 584–596 (2005)
9. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of ACM SIGIR International Conference on Research and Development in Information Retrieval, pp. 50–57 (1999)
10. Joachims, T.: Text Categorization With Support Vector Machines: Learning With Many Relevant Features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
11. Karimzadehgan, M., Zhai, C.: Estimation of Statistical Translation Models Based on Mutual Information for Ad Hoc Information Retrieval. In: Proceedings of ACM SIGIR International Conference on Research and Development in Information Retrieval, pp. 323–330 (2010)
12. McCallum, A., Nigam, K.: A Comparison of Event Models for Naive Bayes Text Classification. In: AAAI 1998 Workshop on Learning for Text Categorization, pp. 41–48 (1998)
13. Nigam, K., Mccallum, A.K., Thrun, S., Mitchell, T.: Text Classification from Labeled and Unlabeled Documents using EM. Machine Learning 39(2/3), 103–134 (2000)
14. Peng, F., Schuurmans, D.: Combining Naive Bayes and n-Gram Language Models for Text Classification. In: Sebastiani, F. (ed.) ECIR 2003. LNCS, vol. 2633, pp. 335–350. Springer, Heidelberg (2003)
15. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Proceedings of ACM SIGIR International Conference on Research and Development in Information Retrieval, pp. 275–281 (1998)
16. Schapire, R.E., Singer, Y.: BoosTexter: A Boosting-based System for Text Categorization. Machine Learning 39(2/3), 135–168 (2000)
17. Schneider, K.-M.: Weighted Average Pointwise Mutual Information for Feature Selection in Text Categorization. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 252–263. Springer, Heidelberg (2005)
18. Wei, X., Croft, W.B.: Modeling Term Associations for Ad-Hoc Retrieval Performance Within Language Modeling Framework. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECIR 2007. LNCS, vol. 4425, pp. 52–63. Springer, Heidelberg (2007)
19. Wu, M.S., Lee, H.S., Wang, H.M.: Exploiting semantic associative information in topic modeling. In: Proceedings of the IEEE Workshop on Spoken Language Technology, pp. 384–388 (2010)
20. Yang, Y.: An Evaluation of Statistical Approaches to Text Categorization. Journal of Information Retrieval 1(1-2), 67–88 (1999)
21. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to Ad Hoc information retrieval. In: Proceedings of ACM SIGIR International Conference on Research and Development in Information Retrieval, pp. 334–342 (2001)
22. Zhou, G., Lua, K.: Interpolation of n-gram and mutual-information based trigger pair language models for Mandarin speech recognition. Computer Speech and Language 13(2), 125–141 (1999)

A Double-Ensemble Approach for Classifying Skewed Data Streams

Chongsheng Zhang[1] and Paolo Soda[2]

[1] School of Computer and Information Engineering, HeNan University, China
chongsheng.zhang@yahoo.com

[2] Integrated Research Centre, Università Campus Bio-Medico di Roma, Italy
p.soda@unicampus.it

Abstract. Nowadays, many applications need to handle large amounts of streaming data, which often presents a skewed distribution, i.e. one or more classes are largely under-represented in comparison to the others. Unfortunately, little effort has been directed towards the classification of skewed data streams, although class-imbalance learning has already been studied in the area of pattern recognition on static data. Furthermore, while existing class-imbalance learning methods increase the recognition accuracy on minority class, they often harm the global classification accuracy. Motivated by these observations, we develop an approach suited for classifying skewed data streams, which integrates two ensembles of classifiers, each one suited for non-skewed and skewed data. This approach substantially increases the global accuracy compared to existing classification methods for skewed data. Experimental tests have been carried out on three public datasets showing interesting results. As a further contribution, we will study metrics to evaluate the performance of skewed data streams classification. We will also review the literature on class-imbalance learning, and skewed data streams classification.

1 Introduction

These days many applications deal with large amounts of transaction data, i.e. network traffic data, sensor network data and web usage data [3]. Such data, also referred to as data streams in the rest of the paper, often present skewed distributions, i.e. some classes are not sufficiently represented while instances of other classes are over-represented.

Class imbalance exists in a large number of real-world domains and, hence, learning on the static imbalanced data has received great focus [4,6]. Existing solutions can be divided into the following four categories: (i) under-sampling the majority class, so that its size matches that of the minority class(es); (ii) over-sampling the minority class so as to match the size of the other class(es); (iii) internally biasing the learning process so as to compensate for class imbalance; (iv) multi-experts systems. Despite such efforts, most of these methods, while increase the accuracy on the minority class, decrease the global accuracy in comparison with traditional learning algorithms.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 254–265, 2012.

Turning our attention to data streams classification, recent research has been directed towards the topic of data streams classification [7,15,8,16]. Few methods, however, have been designed to classify skewed data streams [9].

Therefore, skewed data streams classification deserves more attention. In this respect, we propose here a classification method for skewed data streams, presenting the following contributions: (i) we discuss the pros and cons of metrics for performance evaluation under class skew; (ii) we present a review of the literature concerning classification methods for both static and streaming skewed datasets; (iii) we propose a new approach for skewed data streams classification. Comparing with existing methods, our proposed method improves not only the accuracy on each class but also the global recognition accuracy, as confirmed by experiments carried out on three public datasets.

The rest of the paper is organized as follows: we present background and motivations in section 2, where we also review related work. In section 3, we introduce our approach in detail. In section 4, we report the experimental results. Finally, we conclude the paper in section 5.

2 Background and Motivations

In this paper, we consider two-classes skewed data classification problems, where the minority and majority instances belong to the positive and negative classes respectively, and the positive class is largely under-represented in comparison to the negative one. The skewness of a dataset denotes the degree of data imbalance, and its value is equal to the a priori probability of an instance belonging to the majority class.

2.1 Performance Metrics

For a two-classes classification task, table 1 shows the corresponding confusion matrix which is usually used to assess the performance of a recognition system. We denote $n^- = FP + TN$ and $n^+ = TP + FN$ as the numbers of samples in the negative and positive classes, respectively.

The global recognition accuracy, referred to as *acc*, is a traditional measure for evaluating the performance of a classifier. For a two-classes classification task, $acc = (TP+TN)/(n^- + n^+)$. It is notable that such a measure is sensitive to class skew because it considers values reported in all columns of the confusion matrix. As an example, consider the Credit Card dataset with a skewness of 97.79% (see also subsection 4.1). A classification system would achieve an accuracy as high as $acc = 97.79\%$ if it arbitrarily labels all test samples as negative. However, it would fail to recognize all positive cases, so it cannot meet the need of skewed data classification applications.

As a complementary metric for *acc* on skewed data, we introduce the geometric mean of accuracies (*gacc*) for class-imbalance learning, which is a performance measure used in the literature [12]: $gacc = \sqrt{\prod_{i=1}^{c} \frac{n_{ii}}{n_{+i}}}$, where n_{ii} is the number of elements of class i correctly labeled and n_{+i} is the number of samples

Table 1. Confusion matrix of a two-classes problem

	Actual positive	Actual negative
Hypothesis positive	True Positive (TP)	False Positive (FP)
Hypothesis negative	False Negative (FN)	True Negative (TN)

belonging to class i. Hence, n_{ii}/n_{+i} represents the accuracy for each class. It is clear that *gacc* ranges in $[0, 1]$. For two-classes skewed data classification tasks, we further introduce the following two metrics which specialize in measuring the performance of a classifier on the two different classes:

- *True Positive Rate* or *Recall*, which is defined as $TP_{rate} = acc^{+} = \frac{TP}{TP+FN}$;
- *True Negative Rate*, which is defined as $TN_{rate} = acc^{-} = \frac{TN}{TN+FP}$;

From above definitions, for two classes recognition problem, we obtain $gacc = \sqrt{acc^{+} \cdot acc^{-}}$. On one side, to get a large value of *gacc*, both accuracies should be large. On the other side, *gacc* will be low if either accuracy value is low. Hence, *gacc* is a balance of acc^{+} and acc^{-}. Nevertheless, if we only use the *gacc* value to evaluate a classifier's performance, we can not distinguish its separate performance on the two different classes. As an example, consider the classifier for the Credit Card mentioned above. Its acc^{-} value is 100% but, since its acc^{+} is 0%, the *gacc* value for this classifier is 0%. This example confirms that neither *acc* nor *gacc* on its own is enough to reflect the overall performance of the classifier on skewed data, motivating the use of acc^{+} and acc^{-}.

As a short summary, the metrics of *acc*, *gacc*, acc^{+} (or acc^{-}) should be used together as a joint measure to evaluate classification performance on skewed data streams. Indeed, on the one hand, *acc* measures the global recognition rate and, on the other hand, *gacc* reflects how much classifier performance is balanced. In addition, acc^{+} (or acc^{-}) reports separate classification performance on the two different classes.

2.2 Classification Methods for Skewed Data

Researches for the learning of static imbalanced data can be classified into the following four categories:

1. Under-sampling the majority class by resizing the training sets (**TS**), makes the class distribution more balanced. The main drawback is the removal of the potentially useful samples. One-sided selection is an under-sampling method that tries to overcome tries to overcome this limitation removing borderline and redundant majority class samples, and without touching minority class samples. [2,12].
2. Over-sampling the minority class so as to match the size of the majority one. Synthetic minority over-sampling technique (SMOTE) is an over-sampling approach creating synthetic samples in the feature space along the line segments to join any/all of the k minority class nearest neighbors [5]. Depending

on the amount of needed samples, member samples from the k nearest neighbors are randomly chosen.

3. Internally biasing the discrimination-based process to compensate class imbalance without altering the class distributions. It should assign different weights to prototypes of different classes [13], or use a weighted distance function in the classification phase compensating the TS imbalance without altering the class distribution [1].
4. Multi-experts systems (**MES**). In MES, each composing classifier C_i is trained on a TS composed of a sample subset N_i of the majority class N and all instances from the minority class P. So after sampling a subset N_i from N, C_i is trained on $N_i \cup P$. Later for the test data, the outputs of C_i on the test samples are combined to make the final predication [11]. The main motivation of this approach lies in the observation that a MES generally produces better results than those provided by any of its composing classifiers.

2.3 Classification Methods for Streaming Data

Very fast decision tree learner (VFDT) is an early work for data stream classification [7]. It builds a decision tree incrementally using constant memory. It starts with a single leaf, decides which attribute is the best for splitting the tree, and selects via Hoeffding bound a small subset of examples passing through the nodes. *VFDTc* [8] is an improvement over VFDT that can handle continuous data, incorporate new information online and classify the samples with a single scan of the data.

MES is also applied to data streams building separate classifiers on sequential batches [15]. The performance of existing classifiers are tested using the new batch of data. As a constant number of classifiers is kept, the extra classifiers with worst classification accuracies will be eliminated. The final predication is made by combining the outputs of remaining classifiers through majority voting. In the following, we refer to this method as **SEA**.

Gao et. al. proposed a classification method for skewed data streams [9], which is referred to as **SDM07** in the rest of the paper. To make the class distributions of the TS balanced, they (1) collect minority samples that have appeared over in the new batch and all the past batches, (2) use only the majority instances randomly sampled in the new batch. Samples from steps (1) and (2) are then merged into a new TS used to build a classifier. Moreover, to make more accurate classifications, they generate several such TSs at each new batch by running step (2) several times. The outputs of the set are then combined by majority voting.

2.4 Motivations

The review of the literature reported so far shows that recent research has focused, on the one hand, on class-imbalance learning on static data and, on the other hand, on classifying non-skewed data streams. However, conventional methods for non-skewed data streams usually do not give enough attention to skewed streams, whereas static class-imbalance learning methods often harm the accuracy on majority class, although they increase the recognition accuracy on the minority class.

Table 2. Experimental results on Credit Card dataset using Naïve Bayes

Batch	SEA				SDM07				Our approach			
	acc	*gacc*	acc^+	acc^-	*acc*	*gacc*	acc^+	acc^-	*acc*	*gacc*	acc^+	acc^-
1	0.9084	0.5233	0.2839	0.9644	0.8422	0.6507	0.4843	0.8743	0.8613	0.6551	0.4793	0.8955
2	0.9019	0.5154	0.2773	0.9578	0.8564	0.6335	0.4495	0.8928	0.8731	0.6062	0.4015	0.9154
3	0.9029	0.5074	0.2682	0.9598	0.8634	0.6215	0.4280	0.9024	0.8704	0.6088	0.4065	0.9120
4	0.8942	0.5357	0.3030	0.9472	0.8676	0.6196	0.4230	0.9075	0.8699	0.6152	0.4156	0.9106
5	0.9142	0.4516	0.2086	0.9774	0.8773	0.6143	0.4106	0.9192	0.8803	0.5892	0.3750	0.9256

To better illustrate such motivations, columns 2-9 of Table 2 compare the classification performance achieved by two methods, namely *SDM07* [9] and *SEA* [15], on the Credit Card dataset with skewness of 97.79%. In Table 2, we observe that data streams classification method designed for training under class skew, i.e. *SDM07*, achieves more balanced performance measured in terms of *gacc* than a conventional classifier adopting learning method tailored for non-skewed data, i.e. *SEA*, due to the fact that the minority class classification accuracy (acc^+) of *SDM07* is larger than *SEA*. However, although acc^+ is improved, we observe that *acc* values returned by *SDM07* is always smaller than those provided by *SEA*, confirming our observation that global accuracy decreases for existing methods handling skewed data streams.

Thus, we are motivated to develop a new skewed data streams classification method that can increase both the global recognition accuracy and the accuracy on minority class.

3 Proposed Method

As reported in section 2.4, we have noticed that balancing the accuracies for each class has the side effect of decreasing the global recognition accuracy (*acc*). Therefore, we present in the following a method aiming at achieving larger *acc* while still improving acc^+ or *gacc*.

3.1 Framework of the Method

Since it is very difficult for a class-imbalance learner to achieve high performance on both *acc* and *gacc*, we decide not to pursue perfect performance on both measures separately, but to develop a classification method that can balance them simultaneously, harming the global accuracy less than previous methods.

So, how can we balance *acc* with *gacc*? We utilize a multi-objective optimization technique selecting the final output of the classification system between the output of a classifier trained according to a learning method addressing the course of class imbalance, and the output of a classifier adopting a training method for non-skewed data [14]. This choice is driven by a parameter, referred to as threshold t^* in the following, whose value maximizes two objective functions, i.e. the global accuracy (*acc*) and the geometric mean accuracies (*gacc*), on a validation set.

The framework of our method, shown in Fig. 1 (left), is based on two ensembles of classifiers. They are referred to as non-skewed ensemble of classifiers (**NEC**), and skewed ensemble of classifiers (**SEC**). The former is trained on the original skewed distribution, whereas the latter is trained on an artificially balanced training set, applying MES scheme suited for imbalanced data. In our implementation, we use *SEA* [15] for *NEC*, and *SDM07* [9] for *SEC*. This framework is adapted to data streams by means of dividing the training streams into batches, and each batch is further divided into training and validation sets.

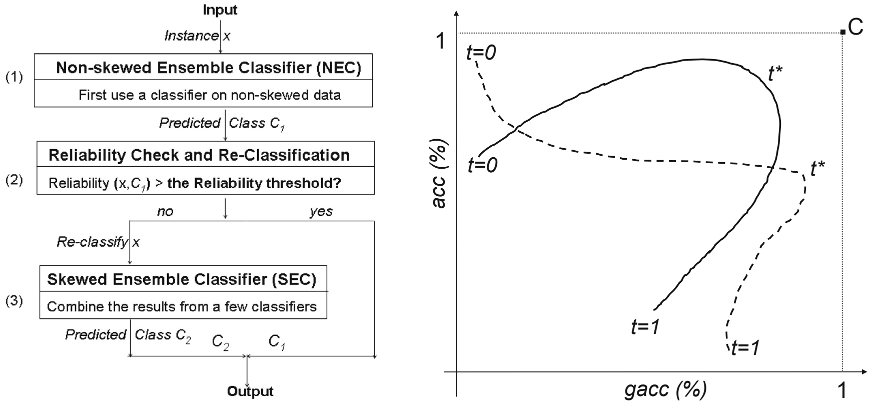

Input
Instance x
(1)
Non-skewed Ensemble Classifier (NEC)
First use a classifier on non-skewed data
Predicted Class C1
Reliability Check and Re-Classification
(2)
Reliability (x,C1) > the Reliability threshold?
no
yes
Re-classify x
Skewed Ensemble Classifier (SEC)
(3)
Combine the results from a few classifiers
Predicted Class C2
C2
C1
Output
1
C
t=0
t*
t=0
t*
acc (%)
t=1
t=1
gacc (%)
1

Fig. 1. Framework of Proposed Method (left) and Example of Acc-Gacc curves (right)

Given an instance x belonging to test set, the final label $O(x)$ is determined as follows:

$$O(x) = \begin{cases} O_{NEC}(x) & \text{if } \phi(x) \geq t^* \\ O_{SEC}(x) & \text{otherwise} \end{cases} \tag{1}$$

When the reliability $\phi(x)$ provided by *NEC* is larger than the threshold t^*, the final label corresponds to the label returned by *NEC* because it is reasonable to assume that *NEC* is likely to provide a correct classification. But, when $\phi(x)$ is below t^*, $O(x)$ is equal to the label assigned by *SEC*, i.e. a classification method tailored specially for skewed data. Indeed, in this case, the value of the reliability suggests that the decision returned by NEC may not be safe. We will explain the rationale of reliability estimation in subsection 3.3.

3.2 Multi-objective Optimization

According to our proposal, we will first train NEC and SEC on the training set of a given batch. Next, both of them are used to classify instances belonging to the validation set of the batch to determine the best value of t^* to be used with the test data. Finally for test data, we apply equation 1 to set the final classification. As reported above, the choice between the outputs of NEC and SEC is driven by t^*. Since t^* is an important threshold parameter, how do we set this parameter? In order to answer this question, recall that *gacc* measures how much the accuracies on two classes are balanced, whereas *acc* estimates the global performance of the classification system. Let us represent *gacc* and *acc* on the X and Y axes respectively, and vary a threshold t to generate a set of points that can be used to plot a curve using samples belonging to validation set. The curve extrema at $t = 0$ and $t = 1$ correspond to *NEC* and *SEC* performance, respectively. In this plot, the ideal point is $\mathbf{C} = (1, 1)$; hence, the nearer the curve to this point, the better the performance obtained. Therefore, the value t^* is given by $\arg\min_t(||\mathbf{p}(t) - \mathbf{C}||)$, where $\mathbf{p}(t)$ is the pair of $gacc(t)$ and $acc(t)$ values measured on the validation set when the threshold t is used.

Fig. 1 (right) shows two examples of this curve, corresponding to two different situations that may occur. The first situation is represented by the continuous line in the figure. In this case, the proposed method selects a value of t^* that permits to improve both *gacc* and *acc* in comparison to individual performance of *NEC* and *SEC*, i.e. points marked with $t = 0$ and $t = 1$. The second situation is represented by the dashed curve. In this case, the proposed method selects a value of t^* that improves *gacc* with respect to both *NEC* and *SEC*, while it reduces slightly the value of *acc* in comparison to *NEC*. We deem that such a reduction can be accepted since final performance are more balanced than individual ones returned by *NEC* and *SEC*.

Algorithm 1 shows the algorithm implementing our proposal presented so far. The training stream is divided into sequential batches (line 1), and each batch is further divided into training and validation sets (line 3-(a)). Using the training set, we train *NEC* and *SEC* (line 3-(b)). Next we compute t^* using a validation set and applying the method given in subsection 3.2, (line3-(c)). As *NEC* and *SEC* are both ensemble of classifiers, we collect the member classifiers in line 3-(d). To classify test instances, we apply step 3-(e) according to equation 1.

3.3 Reliability Estimation

This subsection answers to the following question: what is the reliability and why is it useful in the proposed method?

Utilizing information derived from classifier outputs allows for estimating the reliability of each classification act. Reliability takes into account many issues that influence the achievement of a correct classification, such as the noise affecting the samples domain, and the differences between the objects to be recognized and those used for training the classifiers.

Let $\phi(x)$ denote the reliability of a classification act on any instance x, and the value range within $[0, 1]$. For two-class classifiers, $\phi(x)$ is computed using

Algorithm 1. Algorithm of the proposed method

```
1. Divide the labeled dataset $\mathbf{Z}$ into $n$ batches $D_1, D_2, \ldots, D_n$.
2. Let $\mathbf{Z}_{Te}$ be the test set.
3. For each batch:
   (a) Divide the samples into training and validation sets, denoted by $\mathbf{Z}_{Tr}$, $\mathbf{Z}_{Va}$.
   (b) Train the non-skewed ensemble classifier (NEC) and a skewed ensemble classifier ($SEC$) on $\mathbf{Z}_{Tr}$.
   (c) Find the best threshold $t^*$ s.t. the system achieve the largest values of both $acc$ and $gacc$.
   (d) Collect trained $NEC_i$ and $SEC_i$, with $i = 1, 2, \ldots, n$.
   (e) Apply $NEC_i$ and $SEC_i$ to $\mathbf{Z}_{Te}$, using the following classification rule
       $$O(x) = \begin{cases} O_{NEC}(x) & \text{if } \phi(x) \geq t^* \\ O_{SEC}(x) & \text{otherwise} \end{cases}$$
       where $x$ is a sample, $O_{NEC}(x)$ and $O_{SEC}(x)$ are the outputs provided by $NEC$ and $SEC$, $\phi(x)$ is the reliability of $NEC$, and $O(x)$ is the final label.
```

the difference of predictions on the two different classes. A low value of $\phi(x)$ will suggest that the classification decision made on instance x is not safe since, for instance, it may be a borderline instance or it can be affected by noise in the feature space; while a large value of $\phi(x)$ would suggest that the classifier is more likely to have provided a correct classification [10].

In order to explain the rationale of using the reliability for skewed data classification, let us consider Fig. 2, where we report the experimentally measured distributions of reliability values for test samples labeled by a classifier (i.e. *NEC*) trained on a skewed distribution. On the one hand, when we apply *NEC*, the minority class samples are more likely to receive low reliability values (see the left part of Fig. 2). On the other hand, although low reliability values can also be found for true negative instances, instances with high reliability values are more likely to belong to the majority (negative) class (see the right part of Fig. 2).

In short, there are two main reasons for using reliability estimations on skewed data stream classifiers: (1) applying *NEC*, samples with high reliability values are more likely to belong to negative (majority) class. Hence, we can use reliability values to distinguish between positive and negative instances; (2) *SEC* is trained on artificially balanced training sets, so it should recognize not only positive instances, but also negative ones. Therefore, although instances with low reliability values can contain negative (majority) instances, *SEC* should be able to correctly classify most of them.

4 Experimental Evaluation

In this section, we first describe the datasets used for the experiments. Second, we introduce the experimental protocol and, third, we report the experimental results.

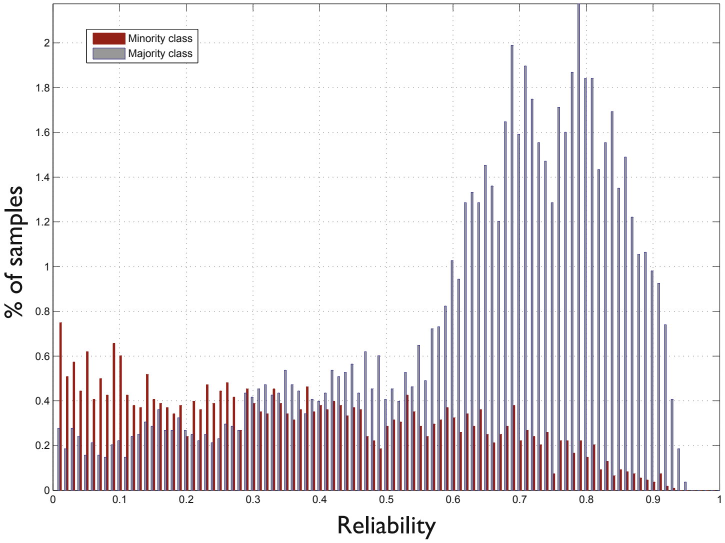


Fig. 2. Examples of reliability distributions for majority and minority class samples

4.1 Datasets

We use the three datasets shown in table 3. These datasets vary in both number of features and skewness. The prediction task for the Adult dataset is to determine whether a person makes over 50K income a year. We only use two classes of the Forest Cover dataset: Ponderosa and Lodgepole Pine. The task of the Forest Cover dataset is to predict the forest cover type. The Credit Card dataset was provided by the 2009 UCSD/FICO data mining contest[1] and used for predicting whether a transaction is an anomaly or not.

Table 3. Datasets description

Datasets	Number of instances	Skewness	Number of features	Source
Adult	44848	70.70%	14	UCI
Forest Cover	319055	88.79%	54	UCI
Credit Card	94682	97.79%	19	UCSD

4.2 Experimental Protocol

We test our approach on the above mentioned datasets. For each dataset, we divide the data into batches, and the last two batches are left for testing only. We vary the size of the batches in our tests: each batch in the Credit Card data contains 10,000 transactions, the size of a batch in the Forest Cover dataset is 20,000, and it is 5,000 for the Adult data.

[1] $http://mill.ucsd.edu$

Table 4. Experimental results on Credit Card dataset using Logistic Regression

Batch	SEA				SDM07				Our approach			
	acc	*gacc*	acc^+	acc^-	*acc*	*gacc*	acc^+	acc^-	*acc*	*gacc*	acc^+	acc^-
1	0.9095	0.5161	0.2757	0.9663	0.7886	0.6490	0.5182	0.8128	0.8692	0.5896	0.3808	0.9130
2	0.9029	0.5469	0.3129	0.9558	0.8027	0.6721	0.5472	0.8256	0.7989	0.6652	0.5381	0.8223
3	0.9082	0.4861	0.2442	0.9677	0.8084	0.6804	0.5571	0.8309	0.8113	0.6725	0.5414	0.8355
4	0.9151	0.5253	0.2839	0.9717	0.8200	0.6729	0.5356	0.8455	0.8740	0.5819	0.3684	0.9193
5	0.9139	0.4941	0.2508	0.9734	0.8084	0.6804	0.5571	0.8309	0.8158	0.6735	0.5397	0.8405
6	0.9084	0.4764	0.2343	0.9688	0.8109	0.6777	0.5505	0.8343	0.8138	0.6658	0.5281	0.8394
7	0.9034	0.4767	0.2359	0.9633	0.8207	0.6844	0.5546	0.8445	0.8664	0.6097	0.4098	0.9073
8	0.9130	0.4625	0.2194	0.9752	0.8147	0.6860	0.5621	0.8373	0.8792	0.5729	0.3543	0.9263

As there are few methods for skewed data stream classification, we implement *SDM07* [9] for *SEC* and we apply *SEA* [15] for *NEC*. These two methods were chosen because both of them are well recognized methods for classifying skewed or non-skewed data streams. Since both *NEC* and *SEC* are classifier ensembles, we use C4.5, Naïve Bayes and Logistic Regression as the base learners in our experiments. Performance are estimated measuring *acc*, *gacc*, acc^+, and acc^-.

4.3 Results

Tables 2, 4 and 5 report the results of the tests we performed on the Credit Card dataset. Tables 6 and 7 show a portion of the test results on both Adult and Forest Cover datasets. It is worth noting that *SEA* usually achieves the largest *acc* value but has the smallest *gacc* value. The case is reversed for *SDM07*, with the largest value for *gacc* but the smallest value for *acc*. Our proposed method, however, achieves a balanced performance between the two above methods. As discussed in section 2, this occurs because *SEA* is a learning method that usually ignores the minority class in skewed data. *SDM07*, on the other hand, is biased toward the minority class but harms the recognition accuracy on majority class. Unlike the other two methods, our proposed approach balances *acc* and *gacc* simultaneously.

We now provide a deeper analysis of the results achieved on the Credit Card dataset (Tables 2, 4 and 5). We notice that: (i) *SEA* usually achieves the best values of *acc*, while *SDM07* often has the best values of both acc^+ and *gacc*; (ii) Sometimes the *gacc* values of our method are as large as or even larger than *SDM07*; (iii) Our method increases the values of the acc^+ of *SEA* by up to 70%. Our method also outperforms *SEA* in terms of *gacc* by 25%; (iv) Our method does not outperform *SEA* in terms of *acc*. With respect to *SEA*, our method decreases *acc* by approximately 4%, but the decrease in *acc* is usually 13% in the case of *SDM07*.

In summary, the above observations show that the proposed method takes into account minority class instances without harming the global accuracy as much as existing methods. We owe this fact to both the double-ensemble framework and the multi-objective optimization technique embedded in the learning algorithm, which dynamically adapts its threshold to variation in data distribution.

Table 5. Experimental results on Credit Card dataset using C4.5

Batch	SDM07				Our approach			
	acc	$gacc$	acc^+	acc^-	acc	$gacc$	acc^+	acc^-
1	0.6908	0.7315	0.7839	0.6825	0.7143	0.7251	0.7384	0.7121
2	0.7261	0.7460	0.7707	0.7221	0.7531	0.7383	0.7210	0.7560
3	0.6952	0.7284	0.7707	0.6884	0.8000	0.6974	0.5944	0.8144
4	0.7153	0.7330	0.7641	0.7109	0.7326	0.7269	0.7202	0.7337
5	0.7101	0.7415	0.7815	0.7307	0.8090	0.6970	0.5681	0.8290
6	0.7163	0.7469	0.7856	0.7100	0.7086	0.7357	0.7699	0.7031
7	0.7124	0.7468	0.7906	0.7054	0.7880	0.7271	0.6614	0.7993
8	0.7100	0.7387	0.7748	0.7042	0.7410	0.7387	0.7359	0.7415

Table 6. Experimental results on Adult dataset using C4.5

Batch	SDM07				Our approach			
	acc	$gacc$	acc^+	acc^-	acc	$gacc$	acc^+	acc^-
1	0.7867	0.8051	0.8438	0.7681	0.7941	0.8092	0.8406	0.7790
2	0.8012	0.8181	0.8535	0.7842	0.8026	0.8136	0.8363	0.7916
3	0.8158	0.8242	0.8411	0.8075	0.8339	0.8021	0.7458	0.8606
4	0.7987	0.8181	0.8589	0.7791	0.8120	0.8087	0.8024	0.8151
5	0.8105	0.8137	0.8201	0.8074	0.8334	0.7841	0.7017	0.8762

Similar results were also found in the experiments with the other two datasets. The results are shown in Table 6 and Table 7.

Table 7. Experimental results on Forest Cover dataset using Naïve Bayes

Batch	SEA				SDM07				Our approach			
	acc	$gacc$	acc^+	acc^-	acc	$gacc$	acc^+	acc^-	acc	$gacc$	acc^+	acc^-
1	0.9430	0.9274	0.9077	0.9475	0.9272	0.9332	0.9411	0.9254	0.9440	0.9262	0.9038	0.9491
2	0.9403	0.9326	0.9228	0.9425	0.9244	0.9360	0.9511	0.9211	0.9393	0.9330	0.9262	0.9398
3	0.9385	0.9308	0.9209	0.9408	0.9249	0.9348	0.9477	0.9220	0.9363	0.9313	0.9250	0.9377
4	0.9403	0.9319	0.9211	0.9428	0.9253	0.9345	0.9465	0.9226	0.9371	0.9289	0.9185	0.9395
5	0.9413	0.9333	0.9232	0.9436	0.9264	0.9354	0.9472	0.9237	0.9403	0.9336	0.9252	0.9422
6	0.9421	0.9313	0.9176	0.9452	0.9262	0.9358	0.9485	0.9233	0.9390	0.9326	0.9244	0.9408
7	0.9379	0.9328	0.9264	0.9393	0.9246	0.9366	0.9525	0.9210	0.9396	0.9318	0.9218	0.9418
8	0.9412	0.9334	0.9235	0.9434	0.9253	0.9363	0.9508	0.9221	0.9369	0.9353	0.9331	0.9374
9	0.9390	0.9319	0.9229	0.9411	0.9239	0.9367	0.9534	0.9202	0.9378	0.9316	0.9237	0.9395
10	0.9396	0.9313	0.9206	0.9420	0.9254	0.9363	0.9506	0.9223	0.9406	0.9303	0.9172	0.9436

Finally, we report the elapsed time during training and test phases of each method. The running time increases with the batch size. Using C4.5 as the base learner on Credit Card data, the proposed method takes 353 seconds, whereas *SDM07* spends 280 seconds. In the case of the Adult data, the proposed method and *SDM07* use 274 and 234 seconds, respectively. These results are reasonable, because the proposed method trains two ensembles of classifiers. Hence, the training time is slight longer than that of *SDM07*.

5 Conclusions

In this paper, we have presented a classification method for skewed data streams. This method is based on two classifier ensembles suited for learning with and without class skew. While still improving the accuracy on each class, the proposed method does not decrease the global recognition accuracy as much as existing methods. Future work will be directed towards extending our study to multi-class data streams.

References

1. Barandela, R., Valdovinos, R.M., Sánchez, J.S.: New applications of ensembles of classifiers. Pattern Analysis & Applications 6(3), 245–256 (2003)
2. Batista, G.E., Carvalho, A.C., Monard, M.C.: Applying One-sided Selection to Unbalanced Datasets. In: Cairó, O., Cantú, F.J. (eds.) MICAI 2000. LNCS, vol. 1793, pp. 315–325. Springer, Heidelberg (2000)
3. Bay, S.D., Kibler, D., Pazzani, M.J., Smyth, P.: The uci kdd archive of large data sets for data mining research and experimentation. SIGKDD Explorations, 81–85 (2000)
4. Chan, P.K., Fan, W., Prodromidis, A.: Distributed data mining in credit card fraud detection. IEEE Intelligent Systems 14, 67–74 (1999)
5. Chawla, N.V., Bowyer, K.W., Hall, L.O.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)
6. Chawla, N.V., Japkowicz, N.: Editorial: Special issue on learning from imbalanced data sets. SIGKDD Explorations 6 (2004)
7. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proc. SIGKDD, pp. 71–80 (2000)
8. Gama, J., Rocha, R., Medas, P.: Accurate decision trees for mining high-speed data streams. In: Proc. SIGKDD, pp. 523–528 (2003)
9. Gao, J., Fan, W., Han, J., Yu, P.S.: A general framework for mining concept-drifting data streams with skewed distributions. In: Proc. SIAM SDM 2007, pp. 3–14 (2007)
10. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Transactions On Pattern Analysis and Machine Intelligence 20(3), 226–239 (1998)
11. Kotsiantis, S., Pintelas, P.: Mixture of expert agents for handling imbalanced data sets. Ann. of Mathematics, Computing and Teleinformatics 1(1), 46–55 (2003)
12. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: One-sided selection. In: Proc. ICML 1997, pp. 179–186 (1997)
13. Pazzani, M., Merz, C., Murphy, P., Ali, K.: Reducing misclassification costs. In: Proc. ICML 1994, pp. 217–225 (1994)
14. Soda, P.: A multi-objective optimisation approach for class imbalance learning. Pattern Recognition 44(8), 1801–1810 (2011)
15. Street, W.N., Kim, Y.: A streaming ensemble algorithm (SEA) for large-scale classification. In: Proc. SIGKDD, pp. 377–382 (2001)
16. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: SIGKDD, pp. 226–235 (2003)

Generating Balanced Classifier-Independent Training Samples from Unlabeled Data

Youngja Park[1], Zijie Qi[2], Suresh N. Chari[1], and Ian M. Molloy[1]

[1] IBM T.J. Watson Research Center, P.O. Box 704,
Yorktown Heights, NY 10598, USA
{young_park,schari,molloyim}@us.ibm.com
[2] University of California Davis, Davis, CA 95616
zqi@ucdavis.edu

Abstract. We consider the problem of generating balanced training samples from an unlabeled data set with an unknown class distribution. While random sampling works well when the data is balanced, it is very ineffective for unbalanced data. Other approaches, such as active learning and cost-sensitive learning, are also suboptimal as they are classifier-dependent, and require misclassification costs and labeled samples. We propose a new strategy for generating training samples which is independent of the underlying class distribution of the data and the classifier that will be trained using the labeled data.

Our methods are iterative and can be seen as variants of active learning, where we use semi-supervised clustering at each iteration to perform biased sampling from the clusters. Several strategies are provided to estimate the underlying class distributions in the clusters and increase the balancedness in the training samples. Experiments with both highly skewed and balanced data from the UCI repository and a private data show that our algorithm produces much more balanced samples than random sampling or uncertainty sampling. Further, our sampling strategy is substantially more efficient than active learning methods. The experiments also validate that, with more balanced training data, classifiers trained with our samples outperform classifiers trained with random sampling or active learning.

1 Introduction

Supervised learning algorithms can provide promising solutions to many real-world problems such as text classification, anomaly detection and information security. A major limitation of supervised learning is the difficulty in obtaining labeled data to train predictive models. Ideally, one would like to train classifiers on diverse labeled data representative of all classes. In many domains, such as text classification or security, there is an abundant amount of unlabeled data, but obtaining a representative subset is challenging: data is typically highly skewed and sparse.

There are two widely used approaches for selecting data to label—random sampling and active learning. Random sampling, a low-cost approach, produces

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 266–281, 2012.

a subset of the data with a similar distribution to the original data set, producing skewed training data for unbalanced data. Training with unbalanced labeled data yields poor results as reported in recent work on the effect of class distribution on learning and performance degradation [1–3]. Active learning produces training data incrementally by identifying the most informative data to label at each phase [4–6]. However, active learning requires knowing the classifier in advance, which is not feasible in many real applications, and requires costly re-training at each step.

In this paper, we present new strategies to generate training samples from unlabeled data to overcome limitations in random and existing active sampling methods. Our core algorithm is an iterative method, in which we generate a small fraction (e.g., 10%) of the desired training set each iteration, independently of *both* the original data distribution as well as the target classifier. More specifically, we first label a small number of randomly selected samples and subsequently apply semi-supervised clustering to embed prior knowledge (i.e., labeled samples) to produce clusters approximating the true classes [7–9]. We then estimate the class distribution of the clusters, and increase the balancedness of the training sample via biased sampling.

A simplistic strategy for biased sampling would be to assume that the class distribution of a cluster is the same as the distribution of labeled samples in the cluster, and to draw samples proportionally to the estimated class distributions. However, this assumption does not hold in early iterations when the number of labeled samples is small, and there is high uncertainty about the class distributions. We present two hybrid approaches to address this issue that perform well in practice. The first approach is to combine the estimated class distribution-based sampling and random sampling. As the number of labeled samples increases, we decrease the influence of random sampling favoring the estimation based on previously labeled samples. The second approach is for cases where additional domain knowledge is available. We use the domain knowledge to estimate class distributions. Domain knowledge may come in many forms, such as conditional probabilities and correlation, e.g., there is a heavy skew in the geographical location of servers hosting malware[10]. We perform a similar transition between the domain knowledge-based density estimation and previously labeled sample-based estimation.

We have validated these strategies on 14 data sets from the UCI data repository [11] as well as a private data set authorizing users to systems (i.e., labeled *grant* and *deny*). These data sets reflect a range of parameters: some are balanced and others highly skewed; and some have binary classes while others have multiple classes. We compare our strategies to random sampling as well as uncertainty based active sampling based on three classifiers: Naive Bayes, Logistic Regression, and SVM. The experiments show that, for highly skewed data sets, our sampling algorithm produces substantially more balanced samples than random sampling. For mildly skewed data sets, our method results in about 25% more minority samples. Similarly, our algorithm performs better than uncertainty sampling based methods for highly skewed samples, producing more than

20% more minority samples on average. For mildly skewed data sets, our algorithm's results are not statistically different from uncertainty sampling based on logistic regression. Given that uncertainty sampling requires one to fix the classifier to be trained and is much slower, we conclude that our algorithm is *always* preferable to both random and uncertainty based sampling. We test the domain knowledge based strategy on the access control permission datasets. Our result show that, in most cases, the addition of domain knowledge significantly improves the convergence of the sampling so we can produce balanced sample sets more quickly.

The quality of training data can best be evaluated by the performance of classifiers trained on this data. We have compared various sampling strategies by training and testing a range of classifiers. Our tests show that the classifiers built with our training data outperform other classifiers in most of the experimental scenarios and produce more consistent performance. Further, our classifiers often outperform uncertainty sampling on AUC and F1 measures, even when sampling and classification used the same classifier. The experimental results confirm that our sampling methods are very generic and can produce highly balanced training data irrespective of the underlying data distribution and the target classifier.

2 Related Work

There is an extensive body of work on generating "good" training data sets. A common approach is active learning, which iteratively selects informative samples, e.g., near the classification border, for human labeling [6, 12–14]. The sampling schemes most widely used in active learning are uncertainty sampling and Query-By-Committee sampling [13, 15, 16]. Uncertainty sampling selects the most informative sample determined by one classification model, while QBC sampling determines informative samples by a majority vote. A major problem with active learning is that the update process is very expensive as it requires classification of all data samples and retraining of the model at each iteration. This cost is prohibitive for large scale problems. Techniques such as batch mode active learning [17, 18] have been proposed to improve the efficiency of uncertainty learning. However, as the batch size grows, the effectiveness of active learning decreases [18–20].

Another approach is re-sampling, i.e., over- and under-sampling classes [21, 22], however this requires labeled data. Recent work combines active learning and re-sampling to address class imbalance in unlabeled data. Tomanek and Hahn [23] propose incorporating a class-specific cost in the framework of QBC-based active learning for named entity recognition. By setting a higher cost for the minority class, this method boosts the committee's disagreement value on the minority class resulting in more minority samples in the training set. Zhu and Hovy [24] incorporate over- and under-sampling in active learning for word sense disambiguation. Their algorithm uses active learning to select samples for human experts to label, and then re-samples this subset. In their experiments, under-sampling caused negative effects but over-sampling helps increase balancedness. However, both [24] and [23] are primarily designed and applied to binary classification problems for text and are hard to generalize to multi-class problems and non-text domains.

Our approach is iterative like active learning, but it differs crucially in that it relies on semi-supervised clustering instead of classification and selects target samples based on estimated class distribution in each cluster. This makes it more general where the best classifier is not known in advance. Ours is the first attempt at using active learning with semi-supervised clustering instead of classification and thus does not suffer from over-fitting. Furthermore, since most classification methods require the presence of at least two different classes in the training set, there is a challenge in providing the initial labeling sample for active learning; an arbitrary insertion of instances from at least two classes is required. Our method does not have this limitation, and, although not shown in the experiments, performs as well with a random initial sample. Our work provides a general framework which is domain independent and can be easily customized to specific domains.

3 Generating Balanced Training Data

Our strategy for generating balanced training sets are described in this section. Section 3.1 presents a high level overview of the algorithm, and later sections provide a more formal description and specific instantiations of the key steps and discuss various tradeoffs.

3.1 Overview

Given an unlabeled dataset with unknown class distribution, potentially skewed, our goal is to produce balanced labeled samples for training predictive models. Formally, we can define the balanced training set problem as follows:

Definition 1. Let $\mathcal{D}$ be an unlabeled data set containing ℓ classes from which we wish to select $\mathcal{T}$, a subset of $\mathcal{D}$ of size N. Let $\mathcal{L}(T)$ be the labels of the training data set T, then the balancedness of $\mathcal{T}$ is defined as the distance between the label distribution of $\mathcal{L}(T)$ and the discrete uniform distribution with ℓ classes, i.e., $D(\mathsf{Uniform}(\ell) \parallel \mathsf{Multi}\,(\mathcal{L}(T)))$. The balanced training set problem is the problem of finding a training data set that minimizes this distance.

If we know the class labels in a given set, then we can use over- and under-sampling to draw balanced sample set [21, 22, 25]. However, the class labels are not known, so instead we must use a series of approximations to approach the results of this ideal solution. We apply an iterative semi-supervised clustering algorithm to estimate the class distribution in the unlabeled data set and guide the sample selection to produce a balanced set. In each iteration, the algorithm draws a batch of samples ($\mathcal{B}$), and domain experts provide the labels of the selected samples. The labeled samples are used in subsequent iterations.

Algorithm 1 is a high level description of our strategy. It takes three inputs: $\mathcal{D}$, an unlabeled data set; ℓ, the number of target classes in $\mathcal{D}$; and N, the number of training samples to generate. We note that the value of the input parameters

```
TrainingSetGeneration($\mathcal{D}, \ell, N, [\mathcal{B}]$)
if $\mathcal{B}$ is undefined then
    $\mathcal{B} \leftarrow \min\left(\left\lfloor \frac{|\mathcal{D}|}{10} \right\rfloor, \frac{N}{10}\right)$ ;
end
$\max_{cluster} \leftarrow \left\lfloor \frac{|\mathcal{D}|}{10} \right\rfloor$ ;
$\mathcal{T} \leftarrow \mathcal{L}$($\mathcal{B}$ randomly selected samples);
while $|\mathcal{T}| < N$ do
    $\{C_1, \ldots, C_k\} \leftarrow$ SemiSupervisedClustering($\mathcal{D}, \mathcal{T}, \max_{cluster}$);
    $\mathcal{T}' \leftarrow \emptyset$;
    foreach $j = 1$ to $k$ do
        $num_j \leftarrow$ DetermineOptimalNumberToSample($C_j$);
        $\mathcal{T}_j \leftarrow$ MaximumEntropySampling($C_j, num_j$);
        $\mathcal{T}' \leftarrow \mathcal{T}' \cup \mathcal{T}_j$;
    end
    $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{L}(\mathcal{T}')$;
end
```

Algorithm 1. High level steps of the proposed algorithm

are previously determined in most applications. The number of samples to select in each iteration, $\mathcal{B}$, can be determined automatically based on $\mathcal{D}$ and N, or users can optionally set the batch size as an input parameter.

To start, we select $\mathcal{B}$ samples randomly and obtain the labels. Then, a semi-supervised clustering algorithm is applied to embed the labels obtained from the prior steps into the clustering process (Section 3.2), which can be used to approximate the class distributions in the clusters. The key intuition behind the process is that we want to extract more samples from clusters which are likely to increase the balancedness of the overall training set. Our algorithm tries to infer the class distribution of each cluster and use this to over- or under-sample. Section 3.3 describes key details of the class density estimation process, the various tradeoffs and their influence on the ultimate results. Once we determine how much to sample from each cluster, we obtain diverse samples using maximum entropy sampling (Section 3.5). We note that there is an implicit secondary optimization of maximizing the entropy of the sampled points, $\mathcal{H}(\mathcal{T})$, which is the byproduct of the real objective, maximizing the performance of a classifier trained on $\mathcal{T}$.

3.2 Semi-supervised Clustering

Semi-supervised clustering is a technique which incorporates existing information into clustering. A number of approaches have been proposed to embed constraints into existing clustering algorithms [9, 26]. We explore two different strategies: a distance metric technique for multi-variate numeric data and a heuristic to add class labels in the feature set for categorical data. We use Relevant Component Analysis (RCA) [7] for distance metric-based semi-supervised clustering, This is a Mahalanobis metric technique which finds a new space with the most relevant features in the side information. It learns a global distance metric parameterized by a transformation matrix $\hat{C}$ to capture relevant features in the labeled sample set. It maximizes the similarity between the original data set X and the new representation Y constrained by the mutual information $I(X, Y)$. By projecting X into the new space through transformation $Y = \hat{C}^{-\frac{1}{2}} X$, two projected data objects, Y_i, Y_j, in the same connected component have a smaller distance.

Here, we sketch the steps to compute the "within-chunklet" covariance matrix (transformation matrix), $\hat{C}$. Given a data set $X = \{x_i\}_{i=1}^N$ and a labeled sample set $L \subset X$, suppose u connected components (i.e., chunklets) $M = \{M_j\}_{j=1}^u$ are obtained based on L, which satisfies $X = \bigcup_{i=1}^u M_j$. Let the data points in a component M_j be denoted as $\{x_{ji}\}_{i=1}^{|M_j|}$ for $1 \leq j \leq u$. Then, the covariance matrix $\hat{C}$ is defined by Equation 1, where m_j is the centroid of M_j.

$$\hat{C} = \frac{1}{N} \sum_{j=1}^{u} \sum_{i=1}^{|M_j|} (x_{ji} - m_j)(x_{ji} - m_j)^T \tag{1}$$

After projecting the data set into a new space using RCA, the data set is recursively partitioned until all the clusters are smaller than a predetermined threshold, $\max_{cluster}$. Algorithm 2 summarizes our semi-supervised clustering algorithm using RCA.

```
**SemiSupervisedClustering**($X, \mathcal{L}(T), \max_{cluster}$)
$\hat{C} = RCA(X, \mathcal{L}(T))$;
$Y = \hat{C}^{-\frac{1}{2}} X$;
$C \leftarrow \{C_1, C_2\} = BinaryClustering(Y)$;
while $\exists C_i \in C, |C_i| > \max_{cluster}$ do
    $\{C_{i1}, C_{i2}\} \leftarrow BinaryClustering(C_i)$;
    $C \leftarrow \big(C \setminus C_i\big) \cup \{C_{i1}, C_{i2}\}$ ;
end
```

Algorithm 2. Semi-supervised clustering algorithm to divide a data set into balanced clusters

The RCA algorithm makes several assumptions regarding the distribution of the data. Primarily, it assumes the data is multivariate normally distributed, and, if so, produces the optimal result. It has also been shown to perform well on datasets when the normally distributed assumption fails [7], including many of the UCI datasets used in this work. However, it is not known to work well for Bernoulli or categorical distributed data, such as the access control datasets, where it produces a marginal improvement, at best. Instead, we choose a simple method by augmenting the feature set with labels of known samples, i.e., $\mathcal{F} \parallel \mathcal{L}$, and assigning a default feature value, or holding out feature values, for unlabeled samples. For example, if we have ℓ class labels, we will add ℓ new binary features. If the sample has class j, we will assign feature j a value of 1, and all other label features a zero. Unlabeled samples are assigned a feature corresponding to the prior, the fraction of labeled samples with that class label. As before, we use the recursive binary clustering technique described previously to cluster the data. We find that this simple heuristic produces good clusters and yields balanced samples more quickly for categorical data.

3.3 Determine the Optimal Number to Samples from Each Cluster

Once we have clustered the data, the key step is to estimate the class density in the clusters and use this information to perform biased sampling to increase the overall balancedness in the sample set. We assume that the semi-supervised clustering step has produced biased clusters allowing us to approximate a solution of drawing samples with known classes.

The first approach is to assume that the class distribution of the cluster is exactly the same as the class distribution of the labeled samples in this cluster. This is based on the optimistic assumption that the semi-supervised clustering works perfectly and groups together elements similar to the labeled sample. First, we determine how many samples we wish to draw from each class in this iteration from the total $\mathcal{B}$ samples to draw. Let ℓ_i^j be the number of instances of class j sampled after iteration i, and ρ_i^j be the normalized proportion of samples with class label j, i.e., $\rho_i^j = \frac{\ell_i^j}{\sum_r \ell_i^r}$. To increase balancedness, we want to sample inverse proportionally to their current distribution [21, 22, 25], i.e., $n_j = \frac{1-\rho_i^j}{\ell-1} * \mathcal{B}$, where ℓ is the number of classes. Next, we use the estimated class distribution in each cluster to determine the appropriate number of samples to draw from each class. Let θ_i^j be the probability of drawing a sample with class label j from the *previously labeled subset* of cluster i. By our assumption, this is exactly the probability of drawing a sample with class label j from the *entire* cluster. To sample n_j samples with label j, we draw $n_j \frac{\theta_i^j}{\sum_{l=1}^k \theta_l^j}$ samples from cluster i, where k is the number of clusters. Another strategy is to draw all n_j samples from the cluster with the maximum probability of drawing class j, however our method selects a more representative subset, and we can obtain good results even if our estimation of cluster densities is incorrect and reduces later classifier over-fitting.

In early stages of the iteration process, where there are few labeled samples, this approach does not work well. We use a hybrid approach where we select a certain percentage of $\mathcal{B}$ samples based on the estimates of class distribution and select remaining samples randomly from all clusters. We increase the influence of labeled samples over time as we obtain more labeled samples and thus better estimates on class distribution. Let $\mathcal{B}_\mathcal{L}$ be the number of samples to select based on labeled samples, and $\mathcal{B}_r$ be the number of samples to select randomly. Then, we compute $\mathcal{B}_\mathcal{L} = \omega \cdot \mathcal{B}$ and $\mathcal{B}_r = (1-\omega) \cdot \mathcal{B}$ using a sigmoid function $\omega = \frac{1}{1+e^{-\lambda t}}$, where t is the iteration number and λ a parameter that controls the rate of mixing. As t increases (i.e., number of labeled samples increases), we decay the influence of random sampling favoring empirical estimates.

3.4 Leveraging Domain Knowledge

In Section 3.3, we presented a hybrid sampling method that combines sampling based on the class distribution of each cluster and random sampling. In many settings, domain experts may have additional knowledge regarding the distribution of class labels and correlations with given features or feature values. For instance, in the problem of detecting malicious web sites, there is a heavy skew in geographical location of the web servers [10]. In access control datasets, one can expect correlations between the department of the employee and granted permissions. This section outlines a method where we can incorporate such domain knowledge to estimate the class distribution within a cluster. We use domain knowledge in the form of a correlation value between a feature and a class label. For example, $\mathsf{corr}(\mathit{Department} = 20, \mathit{class} = \mathit{grant}) = +0.1$. These correlations may be noisy and incomplete, pertaining to only a small number of features or feature values. Without loss of generality, we will only consider binary labels; the technique can readily be extended to non-binary labels.

Given a small number of feature-class and feature-value-class correlations and the feature distribution within a cluster, we can estimate the class density. We leverage some of the ideas from the MYCIN model of inexact reasoning [27]. They note that domain knowledge is often logically inconsistent and non-Bayesian. For example, given expert knowledge that $p\left(\mathit{class} = \mathit{grant} \mid \mathit{Department} = 20\right) = 0.6$, we *cannot* conclude that $p\left(\mathit{class} \neq \mathit{grant} \mid \mathit{Department} = 20\right) = 0.4$. Further, a naïve Bayesian approach requires an estimation of the global class distribution, which we assume is not known a priori. Instead, our approach is based on independently aggregative suggestive evidence and leverages properties from fuzzy logic. The correlations correspond to inference rules (e.g., $\mathit{Department} = 20 \rightarrow \mathit{class} \neq \mathit{grant}$), where the correlation coefficients are the confidence weights of the inference rules, and the feature density within each class is the degree that the given inference rule is fired. We evaluate each inference rule in support (positive correlation) and refuting (negative correlation) the class assignments, and aggregate the results using the Product T-Conorm, $\mathsf{norm}(x, y) = x + y - x * y$. We combine evidence supporting and refuting a class assignment using the rule "class 1 and not class 2," and T-Norm for conjunction, $f(x, y) = x * (1 - y)$. Finally, we use domain knowledge-based estimates to supplement the empirical estimates $\mathcal{B}_{\mathcal{L}}$. Let $\mathcal{B}_d$ be the number of samples to select based on domain knowledge, then we select $\mathcal{B} = \mathcal{B}_{\mathcal{L}} + \mathcal{B}_d$ samples in each iteration. As domain knowledge is inexact and noisy, we decay the influence of its estimates over time, favoring the empirical estimates using the sigmoid ω described in Section 3.3, i.e., $\mathcal{B}_d = (1 - \omega) \cdot \mathcal{B}$.

3.5 Maximum Entropy Sampling

Finally, given the set of clusters $\{C_i\}_{i=1}^{k}$, and the number of samples to select from each cluster, we sample to maximize the entropy of the sample $\mathcal{L}(T)$. We assume that the data in each cluster follows a Gaussian distribution. For a continuous variable $x \in C_i$, let the mean be μ, and the standard deviation be

σ, then the normal distribution $\mathcal{N}(\mu, \sigma^2)$ has maximum entropy among all real-valued distributions. The entropy for a multivariate Gaussian distribution [28] is defined as:

$$\mathcal{H}(X) = \frac{1}{2} d \left(1 + \log(2\pi)\right) + \frac{1}{2} \log\left(|\Sigma|\right) \tag{2}$$

where d is the dimension, Σ the covariance matrix, and $|\Sigma|$ the determinant of Σ. Thus, more variation the covariance matrix has along the principal directions, the more information it embeds.

Note that the number of possible subsets of r elements from a cluster C can grow very large (i.e., $\binom{|C|}{r}$)), so finding a subset with the global maximum entropy can be computationally very intensive. We use a greedy method that selects the next sample which adds the most entropy to the existing labeled set. Our algorithm performs the covariance calculation $O(rn)$ times, while the exhaustive search approach requires $O(n^r)$. If there are no previously labeled samples, we start the selection with the two samples that have the longest distance in the cluster. The maximum entropy-based sampling method is presented in Algorithm 3.

```
**MaximumEntropySampling**($\mathcal{T}, C, num$)
$C_U \leftarrow$ unlabeled samples in $C$;
$\mathcal{T}_C \leftarrow \emptyset$;
**while** $|\mathcal{T}_C| < num$ **do**
    $u \leftarrow \arg\max_{u_i \in C_U} \mathcal{H}(\mathcal{T} \cup \{u_i\})$ ;
    $\mathcal{T}_C \leftarrow \mathcal{T}_C \cup \{u\}$ ;
    $C_U \leftarrow C_U \setminus \{u\}$ ;
**end**
Return $\mathcal{T} \cup \mathcal{T}_C$
```

Algorithm 3. Maximum entropy sampling strategy

4 Experiments and Evaluation

This section presents a performance comparison of our sampling strategies with random sampling and uncertainty based sampling on a diverse collection of data sets. Our results show that our algorithm produces significantly more balanced sets than random sampling in almost all datasets. Our technique also performs much better than uncertainty based sampling for highly skewed sets, and our training samples can be used to train any classifier. We also describe results which confirm the benefits of domain knowledge.

4.1 Evaluation Setup

To evaluate the sampling strategies, we selected 14 data sets from the UCI repository [11] and a private data set containing the assignment of access control permissions to users. The data sets span a wide range of parameters and are

Table 1. Description, size, and distribution of experimental data sets

Data	#Samples	#Classes	Class Distribution
Breast Cancer	699	2	65.52% vs. 34.48%
Ionosphere	351	2	35.90% vs. 64.10%
KDD'99	49,180	2	97.35 vs. 2.65%
Page Blocks	5,028	2	97.71% vs. 2.29%
Pima Indians	768	2	65.10% vs. 34.90%
Breast Cancer (BC) Wisconsin	198	2	76.26% vs. 23.74%
Spambase	4,601	2	60.60% vs. 39.40%
SPECT	267	2	79.40% vs. 20.60%
Iris	150	3	balanced
Wine	178	3	39.89% 26.97% 33.15%
Statlog (Landsat Satellite)	6,435	6	23.82% to 9.73%
Pen Digits	10,992	10	balanced
Poker Hand	25,010	10	Two major (42.4%–50%) & eight minor (4.8%–0.02%)
Letter Recognition	20,000	26	balanced
Access Permission	3,068	2	91.72% vs. 8.28%

summarized in Table 1: some are highly skewed while others are balanced, some are multi-class while others are binary.

All UCI data sets are used unmodified except the *KDD Cup'99* set which contains a "normal" class and 20 different classes of network attacks. In this experiment, we selected only "normal" class and "guess_password" class to create a highly skewed data set. When a data set is provided with a training set and a test set separately (e.g., 'Statlog'), we combined the two sets. The features in the access control data set are typically organization attributes of a user: department name, job roles, whether the employee is a manager, etc. These categorical features are converted to binary features. Since, such access control permissions are assigned based on a combination of attributes, these data sets are also useful to assess the benefits of domain knowledge.

For each data set, we randomly select 80% of the data to be used as unlabeled data, from which training samples are generated. The remaining 20% of the samples is used to test classifiers trained with the training samples. For uncertainty-based active learning, we use three widely used classification algorithms, Naive Bayes, Logistic Regression, and SVM, and these variants are labeled *Un_Naive*, *Un_LR*, and *Un_SVM* respectively. We used the C-support vector classification (C-SVC) SVM with a radial basis function (RBF) kernel, and Logisitc Regression with RBF kernel. All classification experiments were conducted using *RapidMiner*, an open source machine learning tool kit [29]. Logistic Regression in *RapidMiner* only supports binary classification, and thus it was extended to a multi-class classifier using "one-against-all" strategy for multi-class data sets [30]. All experimental results reported here are the average of 10 runs of the experiments.

Table 2. Distance of the sampled class distributions to the uniform distribution. The best performing algorithm for each data set is highlighted in bold.

Distribution	Data	Sample Size	*Random*	*Un_Naive*	*Un_LR*	*Un_SVM*	*Our*
Highly Skewed	Poker Hand	2,000	0.574	0.570	**0.540**	0.557	**0.540**
	Page Blocks	2,000	0.676	0.657	0.646	0.656	**0.642**
	KDD'99	2,000	0.670	0.680	0.291	0.479	**0.282**
	Access Permission	2,000	0.594	0.592	0.535	0.492	**0.472**
Mildly Skewed	Statlog	2,000	0.150	0.247	0.067	**0.049**	0.118
	SPECT	110	0.427	0.419	0.278	**0.212**	0.320
	BC Wisconsin	80	0.361	**0.318**	0.359	0.377	0.324
	Wine	75	0.114	0.184	0.046	0.077	**0.039**
	Breast Cancer	280	0.229	0.190	0.170	0.201	**0.028**
	Pima Indians	310	0.215	0.083	**0.007**	0.056	0.086
	Ionosphere	140	0.208	0.140	**0.061**	0.089	0.076
	Spambase	1,845	0.151	0.199	**0.018**	0.048	0.093
Uniform	Iris	60	0.055	0.335	0.078	0.401	**0.051**
	Letter Recognition	2,000	**0.020**	0.129	0.094	0.137	0.069
	Pendigits	2,000	**0.020**	0.238	0.060	0.064	0.084

4.2 Comparison of Class Distribution in Training Samples

We first evaluate the five sampling methods by comparing the balancedness of the generated training sets. For each run, we continue sampling till the selected training sample contains 50% of the unlabeled samples or we have 2,000 samples, whichever is smaller. The evaluation metrics we use are the balancedness of the training data and the recall of the minority class. As noted above, each run is done with a random 80% of the underlying data set and the results are averaged over 10 runs. We measure the balancedness of a data set as the distance of the sampled class distribution from the uniform class distribution as defined in Definition 2.

Definition 2. *Let X be a data set with k different classes. Then the uniform distribution over X is the probability density function (PDF), $U(X)$, where $U_i = \frac{1}{k}$, for all $i \in k$. Let $P(X)$ be a PDF over the classes produced by a sampling method. Then the balancedness of the sample is defined as the Euclidean distance between the distributions $U(X)$ and $P(X)$* i.e. $d = \sqrt{\sum_{i=1}^{k}(U_i - P_i)^2}$.

Table 2 summarizes the results of balancedness comparison, and Table 3 shows the recall of minority class for all the data sets respectively. Our method produces significantly better results compared to pure random sampling. On *KDD'99*, our sampling algorithm yields 10x more minority samples on average than random. Similarly for *Page Blocks* and the access permission data set, our method produces about 2x more balanced samples. For mildly skewed data sets, our method also produces about 25% more minority samples on the average. For the data sets which are almost balanced, random is the best strategy as expected. Even in this case, our method produces results which are statistically very close to random. Thus, our method is *always* preferable to random sampling.

Table 3. The recall rates for the binary class data sets. *Min. Ratio* refers to the ratio of the minority class in the unlabeled data set. For the access permission data, the average and the standard deviation over multiple data sets are reported.

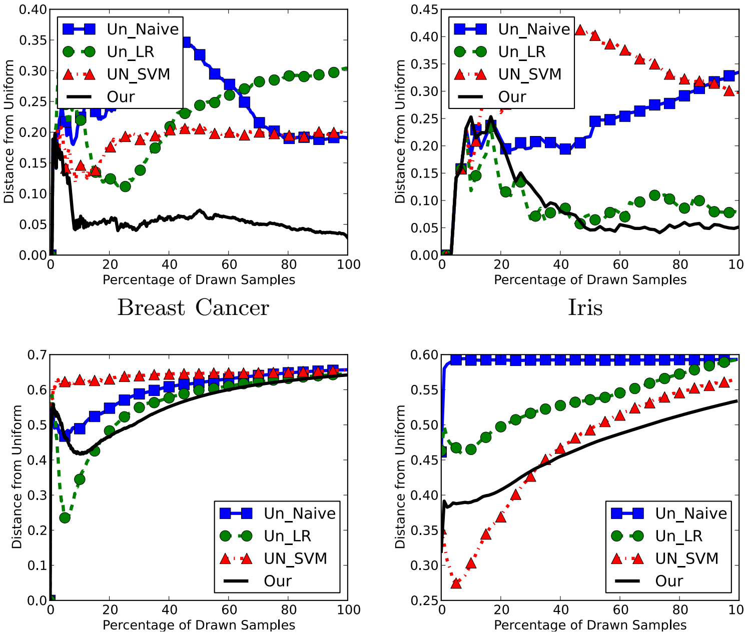
Data	*Min. Ratio*	*Random*	*Un_Naive*	*Un_LR*	*Un_SVM*	*Our*
Poker Hand	0.02	15.00	0.00	0.00	0.00	**62.50**
Page Blocks	2.29	47.61	77.07	93.91	77.83	**100.00**
KDD'99	2.65	5.06	3.63	56.53	30.89	**57.70**
Access Permission	8.04 (5.98)	24.82	25.39 (5.60)	46.32 (22.34)	**60.79 (23.87)**	58.83 (26.31)
Statlog	9.73	39.98	**59.18**	49.84	49.80	35.85
SPECT	20.56	48.18	50.91	75.91	**87.50**	66.59
BC Wisconsin	23.90	51.58	52.89	51.84	49.21	**57.11**
Wine	27.03	46.15	42.05	**60.77**	52.82	57.95
Breast Cancer	34.46	49.07	53.06	41.30	51.97	**75.44**
Pima Indians	34.96	48.56	63.63	**71.77**	64.84	62.28
Ionosphere	35.94	48.88	55.54	**64.95**	60.59	62.28
Spambase	39.41	49.91	45.66	62.00	**67.85**	55.12

Since uncertainty based sampling methods are targeted to cases where the classifier to be trained is known, the right comparison with these methods should include the performance of the resulting classifiers. Further, these algorithms are not very efficient due to re-training at each step. With these caveats, we can directly compare the balancedness of the results. For highly skewed data sets, our method performs better especially when compared to *Un_SVM* and *Un_Naive* methods. On *KDD'99*, we produce 20x and 2x more minority samples compared to *Un_Naive* and *Un_SVM* respectively, while *Un_LR* performs almost as well as our algorithm. Similarly for *Page Blocks*, we perform about 20% better than these methods. We note that our method found all minority samples for all 10 split sets for the *Page Blocks* set. For other data sets, our algorithm shows no significant statistical difference compared to these methods on almost all cases and sometimes we do better. Based on these results, we also conclude that our method is preferable to the uncertainty-based methods based on broader applicability and efficiency.

Figure 1 pictorially depicts the performance of our sampling algorithm as well as the uncertainty based sampling for a few data sets to highlight cases where our method performs better. These figures show the distance from uniform against the percentage of sampled data over iterations. The results show that our sampling technique consistently converges towards balancedness while there is some variation with uncertainty techniques, which remains true for other data sets as well. Note that the distance increases in *Page Blocks* and *Access Permission* data sets after 20% point is because our method exhausted all minority samples.

4.3 Comparison of Classification Performance

In this section, we evaluate the quality of the training samples by comparing the performance of classifiers trained on them. We apply the training samples from the five strategies to train the same type of classifiers (Naive, LR, and SVM), resulting in 15 different "training-evaluation" scenarios. Due to space limitations, we present in Table 4 the AUC and F1-measure for binary class datasets.



Breast Cancer

Iris

Page Blocks

Access Permission

Fig. 1. Progress of balancedness increase over iterations

We expect the performance of the uncertainty sampling methods paired with their respective classifier, e.g., *Un_SVM* with SVM and *Un_LR* with Logistic Regression, to perform best. We observe this behavior on KDD and PIMA, but the off-diagonal entries for uncertainty based sampling show poor results. However, on other datasets such as Breast Cancer and SPECT data sets, our approach outperforms the competing uncertainty sampling. Furthermore, our method performs well consistently across all classifiers without being biased to a single classifier and at reduced computation cost.

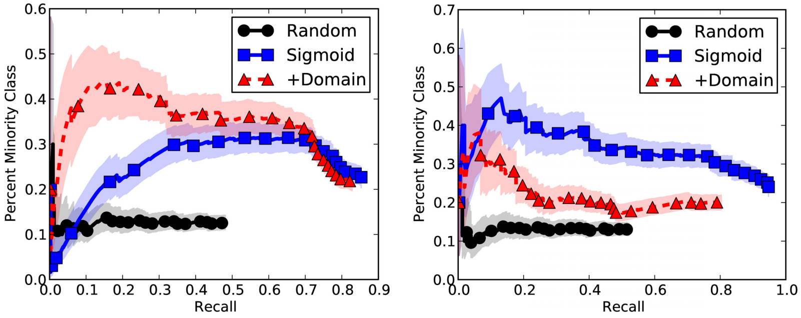
4.4 Impact of Domain Knowledge

The access control permission data sets are used to evaluate the benefit of additional domain knowledge given as a correlation of a user's attributes (e.g., department number, whether she is a manager, etc.) and the granted permission. Our evaluation of sampling with domain knowledge shows that domain knowledge (almost) always helps. There are a few cases where adding domain knowledge negatively impacts performance as shown in Fig 2(b). However, in most cases, domain knowledge substantially improves the convergence of the algorithm. The example depicted in Figure 2(a) is typical of the access control datasets. Since such domain knowledge is mostly used in the early iterations, it significantly helps speed up the convergence.

Table 4. Performance comparison of the three classifiers trained with five different sampling techniques. The figures in bold denote the best performing sampling technique for each classifier. The figures in italics denote the best performing classifier excluding the uncertainty sampling methods paired with their respective classifier.

Data Set	Breast Cancer						SPECT					
Classifier	Naive		LR		SVM		Naive		LR		SVM	
Sampling	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1
Random	0.959	93.52	0.982	91.16	0.549	4.41	***0.796***	50.38	0.713	34.96	0.823	49.12
Un_Naive	0.970	94.04	0.979	76.78	0.541	4.74	0.777	47.97	0.734	43.15	0.812	48.12
Un_LR	0.973	94.06	0.971	61.31	0.524	2.40	0.760	51.29	0.670	45.65	0.818	59.00
Un_SVM	0.961	94.27	0.979	91.17	0.542	2.84	0.787	53.32	0.634	45.65	0.833	53.54
Our	***0.987***	***94.41***	***0.985***	***91.67***	***0.552***	***52.92***	0.766	***54.29***	***0.736***	***45.67***	***0.839***	***56.47***

Data Set	Pima Indians						KDD					
Classifier	Naive		LR		SVM		Naive		LR		SVM	
Sampling	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1
Random	0.785	*60.81*	0.771	11.17	0.658	22.54	0.972	61.94	***0.999***	46.74	0.978	79.91
Un_Naive	**0.808**	**61.14**	0.745	26.53	0.593	28.93	0.964	***63.30***	0.997	21.56	0.954	69.07
Un_LR	0.800	60.10	0.768	**55.93**	0.611	***41.19***	0.905	28.56	**0.999**	**96.81**	***0.998***	*97.37*
Un_SVM	0.785	60.15	0.761	44.02	0.610	31.18	***0.979***	59.24	0.998	*92.74*	0.988	**98.10**
Our	*0.805*	59.5	***0.808***	*51.62*	***0.649***	32.78	0.910	30.54	***0.999***	90.90	0.990	90.68



(a) Positive Impact

(b) Negative Impact

Fig. 2. Comparison of our algorithm ('Sigmoid'), our algorithm with domain knowledge ('+Domain'), and random sampling ('Random'). The y-axis shows the minority class density in the training data, and x-axis shows the recall of the minority class.

5 Conclusion

In this paper, we considered the problem of generating a training set that can optimize the classification accuracy and also is robust to classifier change. We confirmed through experiments that our method produces very balanced training data for highly skewed data sets and outperforms other methods in correctly classifying the minority class. For a balanced multi-class problem, our algorithm outperforms active learning by a large margin and works slightly better than random sampling. Furthermore, our algorithm is much faster compared to active sampling. Therefore, the proposed method can be successfully applied to many real-world applications with highly unbalanced class distribution such as malware detection or fraud detection. In future work, we plan to apply kernel methods for semi-supervised clustering which can discover clusters with nonlinear boundaries in the original space to better fit nonlinearly separable data.

Acknowledgement. This research 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 Agreement Number W911NF-11-C-0200.

References

1. Jo, T., Japkowicz, N.: Class imbalances versus small disjuncts. SIGKDD Explorations 6(1) (2004)
2. Weiss, G., Provost, F.: The effect of class distribution on classifier learning: An empirical study. Dept. of Comp. Science, Rutgers University, Tech. Rep. ML-TR-43 (2001)
3. Zadrozny, B.: Learning and evaluating classifiers under sample selection bias. In: ICML (2004)
4. Dasgupta, S., Hsu, D.: Hierarchical sampling for active learning. In: ICML (2008)
5. Ertekin, S., Huang, J., Bottou, L., Giles, C.L.: Learning on the border: active learning in imbalanced data classification. In: CIKM (2007)
6. Settles, B.: Active learning literature survey. University of Wisconsin-Madison, Tech. Rep. (2009)
7. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning a mahalanobis metric from equivalence constraints. Journal of Machine Learning Research 6 (2005)
8. Wagstaff, K., Cardie, C.: Clustering with instance-level constraints. In: ICML (2000)
9. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side-information. In: Advances in Neural Info. Proc. Systems, vol. 15. MIT Press (2003)
10. Provos, N., Mavrommatis, P., Rajab, M., Monrose, F.: All your iFRAMEs point to us. Google, Tech. Rep. (2008)
11. Frank, A., Asuncion, A.: UCI machine learning repository
12. Campbell, C., Cristianini, N., Smola, A.J.: Query learning with large margin classifiers. In: ICML (2000)
13. Freund, Y., Seung, H.S., Shamir, E., Tishby, N.: Selective sampling using the query by committee algorithm. Machine Learning 28(2-3) (1997)
14. Tong, S., Koller, D.: Support vector machine active learning with application sto text classification. In: ICML (2000)
15. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: SIGIR (1994)
16. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: Computational Learning Theory (1992)
17. Hoi, S.C.H., Jin, R., Zhu, J., Lyu, M.R.: Batch mode active learning and its application to medical image classification. In: ICML (2006)

18. Guo, Y., Schuurmans, D.: Discriminative batch mode active learning. In: NIPS (2007)
19. Schohn, G., Cohn, D.: Less is more: Active learning with support vector machines. In: ICML (2000)
20. Xu, Z., Hogan, C., Bauer, R.: Greedy is not enough: An efficient batch mode active learning algorithm. In: ICDM Workshops (2009)
21. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class imbalance learning. In: IEEE Trans. on Sys. Man. and Cybernetics (2009)
22. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. JAIR 16 (2002)
23. Tomanek, K., Hahn, U.: Reducing class imbalance during active learning for named entity recognition. In: K-CAP (2009)
24. Zhu, J., Hovy, E.: Active learning for word sense disambiguation with methods for dddressing the class imbalance problem. In: EMNLP-CoNLL (2007)
25. wU, Y., Zhang, R., Rudnicky, E.: Data selection for speech recognition. In: ASRU (2007)
26. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: ICML (2001)
27. Shortliffe, E.H., Buchanan, B.G.: A model of inexact reasoning in medicine. Mathematical Biosciences 23(3-4) (1975)
28. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley Interscience (1991)
29. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: Yale: Rapid prototyping for complex data mining tasks. In: Proc. KDD (2006)
30. Rifkin, R.M., Klautau, A.: In defense of one-vs-all classification. J. Machine Learning (1998)

Nyström Approximate Model Selection for LSSVM

Lizhong Ding and Shizhong Liao

School of Computer Science and Technology
Tianjin University, Tianjin 300072, China
szliao@tju.edu.cn

Abstract. Model selection is critical to least squares support vector machine (LSSVM). A major problem of existing model selection approaches is that a standard LSSVM needs to be solved with $O(n^3)$ complexity for each iteration, where n is the number of training examples. In this paper, we propose an approximate approach to model selection of LSSVM. We use Nyström method to approximate a given kernel matrix by a low rank representation of it. With such approximation, we first design an efficient LSSVM algorithm, and then theoretically analyze the effect of kernel matrix approximation on the decision function of LSSVM. Based on the matrix approximation error bound of Nyström method, we derive a model approximation error bound, which is a theoretical guarantee of approximate model selection. We finally present an approximate model selection scheme, whose complexity is lower than existing approaches. Experimental results on benchmark datasets demonstrate the effectiveness of approximate model selection.

Keywords: model selection, Nyström method, matrix approximation, least squares support vector machine.

1 Introduction

Support vector machine (SVM) [18] is a learning system for training linear learning machines in the kernel-induced feature spaces, while controlling the capacity to prevent overfitting by generalization theory. It can be formulated as a quadratic programming problem with linear inequality constraints. The least squares support vector machine (LSSVM) [16] is a least squares version of SVM, which considers equality constraints instead of inequalities for classical SVM. As a result, the solution of LSSVM follows directly from solving a system of linear equations, instead of quadratic programming.

Model selection is an important issue in LSSVM research. It involves the selection of kernel function and associated kernel parameters and the selection of regularization parameter. Typically, the form of kernel function will be determined as several types, such as polynomial kernel and radial basis function (RBF) kernel. In this situation, the selection of kernel function amounts to tuning the kernel parameters. Model selection can be reduced to the selection of kernel parameters and regularization parameter which minimize the expectation of test error [4]. We usually refer to these parameters collectively as *hyperparameters*. Common model selection approaches mainly adopt a nested two-layer inference [11], where the inner layer trains the classifier for fixed hyperparameters and the outer layer tunes the hyperparameters to minimize the generalization

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 282–293, 2012.

error. The generalization error can be estimated either via testing on some unused data (hold-out testing or cross validation) or via a theoretical bound [17,5].

The k-fold cross validation gives an excellent estimate of the generalization error [9] and the extreme form of cross validation, leave-one-out (LOO), provides an almost unbiased estimate of the generalization error [14]. However, the naive model selection strategy based on cross validation, which adopts a grid search in the hyperparameters space, unavoidably brings high computational complexity, since it would train LSSVM for every possible value of the hyperparameters vector. Minimizing the estimate bounds of the generalization error is an alternative to model selection, which is usually realized by the gradient descent techniques. The commonly used estimate bounds include span bound [17] and radius margin bound [5]. Generally, these methods using the estimate bounds reduce the whole hyperparameters space to a search trajectory in the direction of gradient descent, to accelerate the outer layer of model selection, but multiple times of LSSVM training have to be implemented in the inner layer to iteratively attain the minimal value of the estimates. Training LSSVM is equivalent to computing the inverse of a full $n \times n$ matrix, so its complexity is $O(n^3)$, where n is the number of training examples. Therefore, it is prohibitive for the large scale problems to directly train LSSVM for every hyperparameters vector on the search trajectory. Consequently, efficient model selection approaches via the acceleration of the inner computation are imperative.

As pointed out in [5,3], the model selection criterion is not required to be an unbiased estimate of the generalization error, instead the primary requirement is merely for the minimum of the model selection criterion to provide a reliable indication of the minimum of the generalization error in hyperparameters space. We argue that it is sufficient to calculate an approximate criterion that can discriminate the optimal hyperparameters from the candidates. Such considerations drive the proposal of approximate model selection approach for LSSVM.

Since the high computational cost for calculating the inverse of a kernel matrix is a major problem of LSSVM, we consider to approximate a kernel matrix by a "nice" matrix with a lower computational cost when calculating its inverse. The Nyström method is an effective technique for generating a low rank approximation for the given kernel matrix [19,13,8]. Using the low rank approximation, we design an efficient algorithm for solving LSSVM, whose complexity is lower than $O(n^3)$. We further derive a model approximation error bound to measure the effect of Nyström approximation on the decision function of LSSVM. Finally, we present an efficient approximate model selection scheme. It conforms to the two-layer iterative procedure, but the inner computation has been realized more efficiently. By rigorous experiments on several benchmark datasets, we show that approximate model selection can significantly improve the efficiency of model selection, and meanwhile guarantee low generalization error.

The rest of the paper is organized as follows. In Section 2, we give a brief introduction of LSSVM and a reformulation of it. In Section 3, we present an efficient algorithm for solving LSSVM. In Section 4, we analyze the effect of Nyström approximation on the decision function of LSSVM. In Section 5, we present an approximate model selection scheme for LSSVM. In Section 6, we report experimental results. The last section gives the conclusion.

2 Least Squares Support Vector Machine

We use $\mathcal{X}$ to denote the input space and $\mathcal{Y}$ the output domain. Usually we will have $\mathcal{X} \subseteq \mathbb{R}^d$, $\mathcal{Y} = \{-1, 1\}$ for binary classification. The training set is denoted by

$$\mathcal{S} = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)) \in (\mathcal{X} \times \mathcal{Y})^n .$$

We seek to construct a linear classifier, $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \phi(\boldsymbol{x}) + b$, in a feature space $\mathcal{F}$, defined by a feature mapping of the input space, $\phi : \mathcal{X} \to \mathcal{F}$. The parameters $(\boldsymbol{w}, b)$ of the linear classifier are given by the minimizer of a regularized least-squares training function

$$L = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2\mu}\sum_{i=1}^{n}[y_i - \boldsymbol{w} \cdot \phi(\boldsymbol{x}_i) - b]^2, \tag{1}$$

where $\mu > 0$ is called regularization parameter. The basic training algorithm for LSSVM [16] views the regularized loss function (1) as a constrained minimization problem

$$\begin{aligned} \min \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2\mu}\sum_{i=1}^{n}\varepsilon_i^2, \\ \text{s.t.} \quad & \varepsilon_i = y_i - \boldsymbol{w} \cdot \phi(\boldsymbol{x}_i) - b. \end{aligned} \tag{2}$$

Further, we can obtain the dual form of Equation (2) as follows

$$\sum_{j=1}^{n}\alpha_j\phi(\boldsymbol{x}_j) \cdot \phi(\boldsymbol{x}_i) + b + \mu\alpha_i = y_i, \quad i = 1, 2, \ldots, n, \tag{3}$$

where $\sum_{i=1}^{n}\alpha_i = 0$. Noting that $\phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j)$ corresponds to the kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, we can write Equation (3) in a matrix form

$$\begin{bmatrix} \boldsymbol{K} + \mu\boldsymbol{I}_n & \boldsymbol{1} \\ \boldsymbol{1}^{\mathrm{T}} & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ 0 \end{bmatrix}, \tag{4}$$

where $\boldsymbol{K} = [K(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1}^{n}$, $\boldsymbol{I}_n$ is the $n \times n$ identity matrix, $\boldsymbol{1}$ is a column vector of n ones, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^{\mathrm{T}} \in \mathbb{R}^n$ is a vector of Lagrange multipliers, and $\boldsymbol{y} \in \mathcal{Y}^n$ is the label vector.

If we let $\boldsymbol{K}_{\mu,n} = \boldsymbol{K} + \mu\boldsymbol{I}_n$, we can write the first row of Equation (4) as

$$\boldsymbol{K}_{\mu,n}(\boldsymbol{\alpha} + \boldsymbol{K}_{\mu,n}^{-1}\boldsymbol{1}b) = \boldsymbol{y}. \tag{5}$$

Therefore, $\boldsymbol{\alpha} = \boldsymbol{K}_{\mu,n}^{-1}(\boldsymbol{y} - \boldsymbol{1}b)$. Replacing $\boldsymbol{\alpha}$ with $\boldsymbol{K}_{\mu,n}^{-1}(\boldsymbol{y} - \boldsymbol{1}b)$ in the second row of Equation (4), we can obtain

$$\boldsymbol{1}^{\mathrm{T}}\boldsymbol{K}_{\mu,n}^{-1}\boldsymbol{1}b = \boldsymbol{1}^{\mathrm{T}}\boldsymbol{K}_{\mu,n}^{-1}\boldsymbol{y}. \tag{6}$$

The system of linear equations (4) can then be rewritten as

$$\begin{bmatrix} \boldsymbol{K}_{\mu,n} & \boldsymbol{0} \\ \boldsymbol{0}^{\mathrm{T}} & \boldsymbol{1}^{\mathrm{T}}\boldsymbol{K}_{\mu,n}^{-1}\boldsymbol{1} \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha} + \boldsymbol{K}_{\mu,n}^{-1}\boldsymbol{1}b \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{1}^{\mathrm{T}}\boldsymbol{K}_{\mu,n}^{-1}\boldsymbol{y} \end{bmatrix}. \tag{7}$$

Since $\boldsymbol{K}_{\mu,n} = \boldsymbol{K} + \mu \boldsymbol{I}_n$ is positive definite, the inverse of $\boldsymbol{K}_{\mu,n}$ exists.

Equation (7) can be solved as follows: we first solve

$$\boldsymbol{K}_{\mu,n}\boldsymbol{\rho} = \mathbf{1} \qquad \text{and} \qquad \boldsymbol{K}_{\mu,n}\boldsymbol{\nu} = \boldsymbol{y}. \tag{8}$$

The solution $(\boldsymbol{\alpha}, b)$ of Equation (4) are then given by

$$b = \frac{\mathbf{1}^{\mathrm{T}}\boldsymbol{\nu}}{\mathbf{1}^{\mathrm{T}}\boldsymbol{\rho}} \qquad \text{and} \qquad \boldsymbol{\alpha} = \boldsymbol{\nu} - \boldsymbol{\rho} b. \tag{9}$$

The decision function of LSSVM can be written as $f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b$.

If Equation (8) is solved, we can easily obtain the solution of LSSVM. However, the complexity of calculating the inverse of the matrix $\boldsymbol{K}_{\mu,n}$ is $O(n^3)$. In the following, we will demonstrate that Nyström method can be used to speed up this process.

3 Approximating LSSVM Using Nyström Method

We first introduce a fundamental result of matrix computations [10]: for any matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and positive integer k, there exists a matrix $\boldsymbol{A}_k$ such that

$$\|\boldsymbol{A} - \boldsymbol{A}_k\|_{\xi} = \min_{\boldsymbol{D} \in \mathbb{R}^{m \times n} : \operatorname{rank}(\boldsymbol{D}) \leq k} \|\boldsymbol{A} - \boldsymbol{D}\|_{\xi}$$

for $\xi = \mathrm{F}, 2$. $\|\cdot\|_{\mathrm{F}}$ and $\|\cdot\|_2$ denote the Frobenius norm and the spectral norm. Such $\boldsymbol{A}_k$ is called the optimal rank k approximation of the matrix $\boldsymbol{A}$. It can be computed through the singular value decomposition (SVD) of $\boldsymbol{A}$. If $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is symmetric positive semi-definite (SPSD), $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^{\mathrm{T}}$, where $\boldsymbol{U}$ is a unitary matrix and $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$ is a real diagonal matrix with $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$. For $k \leq \operatorname{rank}(\boldsymbol{A})$, $\boldsymbol{A}_k = \sum_{i=1}^{k} \sigma_i \boldsymbol{U}^i \boldsymbol{U}^{i\mathrm{T}}$, where $\boldsymbol{U}^i$ is the ith column of $\boldsymbol{U}$.

We now briefly review the Nyström method [8,19]. Let $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ be an SPSD matrix. The Nyström method generates a low rank approximation of $\boldsymbol{K}$ using a subset of the columns of the matrix. Suppose we randomly sample c columns of $\boldsymbol{K}$ uniformly without replacement. Let $\boldsymbol{C}$ denote the $n \times c$ matrix formed by theses columns. Let $\boldsymbol{W}$ be the $c \times c$ matrix consisting of the intersection of these c columns with the corresponding c rows of $\boldsymbol{K}$. Without loss of generality, we can rearrange the columns and rows of $\boldsymbol{K}$ based on this sampling such that:

$$\boldsymbol{K} = \begin{pmatrix} \boldsymbol{W} & \boldsymbol{K}_{21}^{\mathrm{T}} \\ \boldsymbol{K}_{21} & \boldsymbol{K}_{22} \end{pmatrix}, \qquad \boldsymbol{C} = \begin{pmatrix} \boldsymbol{W} \\ \boldsymbol{K}_{21} \end{pmatrix}. \tag{10}$$

Since $\boldsymbol{K}$ is SPSD, $\boldsymbol{W}$ is also SPSD. The Nyström method uses $\boldsymbol{W}$ and $\boldsymbol{C}$ from Equation (10) to construct a rank k approximation $\widetilde{\boldsymbol{K}}$ of $\boldsymbol{K}$ for $k \leq c$ defined by:

$$\widetilde{\boldsymbol{K}} = \boldsymbol{C}\boldsymbol{W}_k^{+}\boldsymbol{C}^{\mathrm{T}} \approx \boldsymbol{K}, \tag{11}$$

where $\boldsymbol{W}_k$ is the optimal rank k approximation to $\boldsymbol{W}$ and $\boldsymbol{W}_k^{+}$ is the Moore-Penrose generalized inverse of $\boldsymbol{W}_k$. Since $\boldsymbol{W}$ is SPSD, $\boldsymbol{W}_k = \sum_{i=1}^{k} \sigma_i \boldsymbol{U}^i \boldsymbol{U}^{i\mathrm{T}}$ and therefore $\boldsymbol{W}_k^{+} = \sum_{i=1}^{k} \sigma_i^{-1} \boldsymbol{U}^i \boldsymbol{U}^{i\mathrm{T}}$ for $k \leq \operatorname{rank}(\boldsymbol{W})$.

If we write the SVD of $\boldsymbol{W}$ as $\boldsymbol{W} = \boldsymbol{U}_W \boldsymbol{\Sigma}_W \boldsymbol{U}_W^{\mathrm{T}}$, then

$$\boldsymbol{W}_k^+ = \boldsymbol{U}_{W,k} \boldsymbol{\Sigma}_{W,k}^+ \boldsymbol{U}_{W,k}^{\mathrm{T}}, \tag{12}$$

where $\Sigma_{W,k}$ and $\boldsymbol{U}_{W,k}$ correspond the top k singular values and singular vectors of $\boldsymbol{W}$. The diagonal elements of $\Sigma_{W,k}$ are all positive, since $\boldsymbol{W}$ is SPSD and $k \leq \mathrm{rank}(\boldsymbol{W})$.

If we plug Equation (12) into Equation (11), we can obtain

$$\begin{aligned} \widetilde{\boldsymbol{K}} &= \boldsymbol{C}\boldsymbol{U}_{W,k} \boldsymbol{\Sigma}_{W,k}^+ \boldsymbol{U}_{W,k}^{\mathrm{T}} \boldsymbol{C}^{\mathrm{T}} \\ &= \underbrace{\boldsymbol{C}\boldsymbol{U}_{W,k} \sqrt{\boldsymbol{\Sigma}_{W,k}^+}}_{\boldsymbol{V}} \underbrace{\left(\boldsymbol{C}\boldsymbol{U}_{W,k} \sqrt{\boldsymbol{\Sigma}_{W,k}^+}\right)^{\mathrm{T}}}_{\boldsymbol{V}^{\mathrm{T}}}, \end{aligned} \tag{13}$$

where we let $\boldsymbol{V} := \boldsymbol{C}\boldsymbol{U}_{W,k} \sqrt{\boldsymbol{\Sigma}_{W,k}^+} \in \mathbb{R}^{n \times k}$.

For LSSVM, we need to solve the inverse of $\boldsymbol{K} + \mu \boldsymbol{I}_n$. To reduce the computational cost, we intend to use the inverse of $\widetilde{\boldsymbol{K}} + \mu \boldsymbol{I}_n$ as an approximation of the inverse of $\boldsymbol{K} + \mu \boldsymbol{I}_n$. Since $\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}$ is positive semi-definite, the invertibility of $\widetilde{\boldsymbol{K}} + \mu \boldsymbol{I}_{\boldsymbol{n}}$ is guaranteed.

To efficiently calculate the inverse of $\widetilde{\boldsymbol{K}} + \mu \boldsymbol{I}_n$, we further introduce the Woodbury formula [12]

$$(\boldsymbol{A} + \boldsymbol{X}\boldsymbol{Y}\boldsymbol{Z})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{X}\left(\boldsymbol{Y}^{-1} + \boldsymbol{Z}\boldsymbol{A}^{-1}\boldsymbol{X}\right)^{-1} \boldsymbol{Z}\boldsymbol{A}^{-1}, \tag{14}$$

where $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, $\boldsymbol{X} \in \mathbb{R}^{n \times k}$, $\boldsymbol{Y} \in \mathbb{R}^{k \times k}$ and $\boldsymbol{Z} \in \mathbb{R}^{k \times n}$.

Now, we can obtain

$$\begin{aligned} &(\mu \boldsymbol{I}_n + \boldsymbol{K})^{-1} \\ \approx &\left(\mu \boldsymbol{I}_n + \boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}\right)^{-1} \\ = &\frac{1}{\mu}\left(\boldsymbol{I}_n - \boldsymbol{V}\left(\mu \boldsymbol{I}_k + \boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}\right)^{-1} \boldsymbol{V}^{\mathrm{T}}\right). \end{aligned} \tag{15}$$

The last equality of Equation (15) is directly derived from the Woodbury formula with $\boldsymbol{A} = \mu \boldsymbol{I}_n$, $\boldsymbol{X} = \boldsymbol{V}$, $\boldsymbol{Y} = \boldsymbol{I}_k$ and $\boldsymbol{Z} = \boldsymbol{V}^{\mathrm{T}}$.

The essential step of solving LSSVM is to solve Equation (8). If we let $\boldsymbol{u} = [\boldsymbol{\rho}, \ \boldsymbol{\nu}]$ and $\boldsymbol{z} = [\mathbf{1}, \ \boldsymbol{y}]$, Equation (8) is equivalent to

$$(\mu \boldsymbol{I}_n + \boldsymbol{K})\, \boldsymbol{u} = \boldsymbol{z}.$$

Using Equation (15) to replace $\mu \boldsymbol{I}_n + \boldsymbol{K}$ with $\mu \boldsymbol{I}_n + \widetilde{\boldsymbol{K}}$, we can obtain

$$\boldsymbol{u} = \frac{1}{\mu}\left(\boldsymbol{z} - \boldsymbol{V}\left(\mu \boldsymbol{I}_k + \boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}\right)^{-1} \boldsymbol{V}^{\mathrm{T}}\boldsymbol{z}\right). \tag{16}$$

We further introduce a temporary variable $\boldsymbol{t}$ to efficiently solve Equation (16):

$$\begin{aligned} \boldsymbol{t} :\ \left(\mu \boldsymbol{I}_k + \boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}\right) \boldsymbol{t} &= \boldsymbol{V}^{\mathrm{T}}\boldsymbol{z}, \\ \boldsymbol{u} &= \frac{1}{\mu}(\boldsymbol{z} - \boldsymbol{V}\boldsymbol{t}). \end{aligned} \tag{17}$$

We now present an algorithm of solving LSSVM (Algorithm 1).

We estimate the computational complexity of Algorithm 1 in Theorem 1.

Algorithm 1. Approximating LSSVM using Nyström method

```
Input: $n \times n$ kernel matrix $\boldsymbol{K}$, label vector $\boldsymbol{y}$, $c < n$, $k < c$, $\mu$;
Output: $(\alpha, b)$;
1: Calculate $\boldsymbol{C}$, $\boldsymbol{U}_{W,k}$ and $\boldsymbol{\Sigma}^{+}_{W,k}$ according to (10) and (12) using Nyström method;
2: Calculate $\boldsymbol{V} = \boldsymbol{C}\boldsymbol{U}_{W,k}\sqrt{\boldsymbol{\Sigma}^{+}_{W,k}}$ according to (13);
3: Let $\boldsymbol{z} = [\mathbf{1}, \boldsymbol{y}]$ and solve the linear system $\left(\mu \boldsymbol{I}_k + \boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}\right)\boldsymbol{t} = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{z}$ to obtain $\boldsymbol{t}$;
4: Calculate $\boldsymbol{u} = \frac{1}{\mu}(\boldsymbol{z} - \boldsymbol{V}\boldsymbol{t})$ and let $\boldsymbol{\rho}$, $\boldsymbol{\nu}$ be the first and second column of $\boldsymbol{u}$;
5: Calculate $b = \frac{\mathbf{1}^{\mathrm{T}}\boldsymbol{\nu}}{\mathbf{1}^{\mathrm{T}}\boldsymbol{\rho}}$ and $\boldsymbol{\alpha} = \boldsymbol{\nu} - \boldsymbol{\rho} b$ according to (9);
return $(\alpha, b)$;
```

Theorem 1. *The computational complexity of Algorithm 1 is $O(c^3 + nck)$.*

Proof. The computational complexity of step 1 is $O(c^3)$, since the main computational part of this step is the SVD on $\boldsymbol{W}$. In step 2, matrix multiplications are required, so its complexity is $O(kcn)$. In step 3, the inverse of $\left(\mu \boldsymbol{I}_k + \boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}\right)$ is solved by computing Cholesky factorization of it with the complexity $O(k^3)$. The complexity of $\boldsymbol{V}^{\mathrm{T}}\boldsymbol{z}$ is $O(nk)$. The last matrix multiplication to obtain $\boldsymbol{t}$ requires $O(k^2)$. Therefore the total complexity of step 3 is $O(k^3 + nk)$. The complexity of step 4 is $O(nk)$. The complexity of step 5 is $O(n)$, since the multiplication and subtraction between two vectors need to be done. For Nyström approximation, we have $k < c < n$, so the total complexity of Algorithm 1 is $O(c^3 + nck)$. For large scale problems, we usually set $c \ll n$.

Compared to Related Work. Theorem 1 shows that if Nyström approximation is given, we can solve LSSVM in $O(k^3)$. Williams et al. [19] used Nyström method to speed up Gaussian Process (GP) regression. After Nyström approximation was given, they solved GP regression with $O(nk^2)$ complexity. Cortes et al. [6] scaled kernel ridge regression (KRR) using Nyström method. The complexity of their method is $O(n^2c)$ with Nyström approximation (Section 3.3 of [6]).

4 Error Analysis

In this section, we analyze the effect of Nyström approximation on the decision function of LSSVM.

We assume that approximation is only used in training. At testing time the true kernel function is used. This scenario has been considered by [6]. The decision function f derived with the exact kernel matrix K is defined by

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b = \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{k}_x \\ 1 \end{bmatrix},$$

where $\boldsymbol{k}_x = (K(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, K(\boldsymbol{x}, \boldsymbol{x}_n))^{\mathrm{T}}$. We define $\kappa > 0$ such that $K(\boldsymbol{x}, \boldsymbol{x}) \leq \kappa$ and $\widetilde{K}(\boldsymbol{x}, \boldsymbol{x}) \leq \kappa$.

We first consider the effect of Nyström approximation on $\boldsymbol{\rho}$ of Equation (8). Let $\boldsymbol{\rho}'$ denote the solution of $(\widetilde{\boldsymbol{K}} + \mu \boldsymbol{I}_n)\boldsymbol{\rho}' = \mathbf{1}$. We can write

$$\begin{aligned} \boldsymbol{\rho}' - \boldsymbol{\rho} &= (\widetilde{\boldsymbol{K}} + \mu \boldsymbol{I}_n)^{-1}\mathbf{1} - (\boldsymbol{K} + \mu \boldsymbol{I}_n)^{-1}\mathbf{1} \\ &= -\left[(\widetilde{\boldsymbol{K}} + \mu \boldsymbol{I}_n)^{-1}(\widetilde{\boldsymbol{K}} - \boldsymbol{K})(\boldsymbol{K} + \mu \boldsymbol{I}_n)^{-1}\right]\mathbf{1}. \end{aligned} \tag{18}$$

For last equality, we used the identity $\boldsymbol{A}^{-1} - \boldsymbol{B}^{-1} = -\boldsymbol{A}^{-1}(\boldsymbol{A} - \boldsymbol{B})\boldsymbol{B}^{-1}$ for any two invertible matrices $\boldsymbol{A}, \boldsymbol{B}$. Thus, $\|\boldsymbol{\rho}' - \boldsymbol{\rho}\|_2$ can be bounded as follows:

$$\begin{aligned} \|\boldsymbol{\rho}' - \boldsymbol{\rho}\|_2 &\leq \|(\widetilde{\boldsymbol{K}} + \mu \boldsymbol{I}_n)^{-1}\|_2 \, \|\widetilde{\boldsymbol{K}} - \boldsymbol{K}\|_2 \, \|(\boldsymbol{K} + \mu \boldsymbol{I}_n)^{-1}\|_2 \, \|\mathbf{1}\|_2 \\ &\leq \frac{\|\mathbf{1}\|_2}{\mu^2}\|\widetilde{\boldsymbol{K}} - \boldsymbol{K}\|_2 = \frac{\sqrt{n}}{\mu^2}\|\widetilde{\boldsymbol{K}} - \boldsymbol{K}\|_2. \end{aligned} \tag{19}$$

Since $\widetilde{\boldsymbol{K}}$ and $\boldsymbol{K}$ are positive semi-definite matrices, the eigenvalues of $\widetilde{\boldsymbol{K}}+\mu\boldsymbol{I}_n$ and $\boldsymbol{K}+\mu\boldsymbol{I}_n$ are larger than or equal to μ. Therefore the eigenvalues of $(\widetilde{\boldsymbol{K}} + \mu \boldsymbol{I}_n)^{-1}$ and $(\boldsymbol{K} + \mu \boldsymbol{I}_n)^{-1}$ are less than or equal to $1/\mu$.

We further consider $\boldsymbol{\nu}$ of Equation (8). Replacing $\mathbf{1}$ with $\boldsymbol{y}$, we can obtain the similar bound

$$\|\boldsymbol{\nu}' - \boldsymbol{\nu}\|_2 \leq \frac{\|\boldsymbol{y}\|_2}{\mu^2}\|\widetilde{\boldsymbol{K}} - \boldsymbol{K}\|_2 = \frac{\sqrt{n}}{\mu^2}\|\widetilde{\boldsymbol{K}} - \boldsymbol{K}\|_2. \tag{20}$$

As the assumptions, we use the true kernel function at testing time, so no approximation affects $\boldsymbol{k}_x$. For simplicity, we assume the offset b to be a constant ζ. Therefore, the approximate decision function f' is given by $f'(\boldsymbol{x}) = [\boldsymbol{\alpha}'; \zeta]^{\mathrm{T}}[\boldsymbol{k}_x; 1]$.

We can obtain

$$f'(\boldsymbol{x}) - f(\boldsymbol{x}) = \left(\begin{bmatrix}\boldsymbol{\alpha}' \\ \zeta\end{bmatrix}^{\mathrm{T}} - \begin{bmatrix}\boldsymbol{\alpha} \\ \zeta\end{bmatrix}^{\mathrm{T}}\right)\begin{bmatrix}\boldsymbol{k}_x \\ 1\end{bmatrix} = \begin{bmatrix}\boldsymbol{\alpha}' - \boldsymbol{\alpha} \\ 0\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{k}_x \\ 1\end{bmatrix} = (\boldsymbol{\alpha}' - \boldsymbol{\alpha})^{\mathrm{T}}\boldsymbol{k}_x. \tag{21}$$

By Schwarz inequality,

$$|f'(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \|\boldsymbol{\alpha}' - \boldsymbol{\alpha}\|_2\|\boldsymbol{k}_x\|_2 = \sqrt{n}\kappa\|\boldsymbol{\alpha}' - \boldsymbol{\alpha}\|_2. \tag{22}$$

From Equation (9), we know that $\boldsymbol{\alpha} = \boldsymbol{\nu} - \boldsymbol{\rho} b = \boldsymbol{\nu} - \boldsymbol{\rho}\zeta$, so

$$\begin{aligned} \|\boldsymbol{\alpha}' - \boldsymbol{\alpha}\|_2 &\leq \|\boldsymbol{\nu}' - \boldsymbol{\nu}\|_2 + \zeta\|\boldsymbol{\rho} - \boldsymbol{\rho}'\|_2 \\ &\leq \frac{\sqrt{n}}{\mu^2}\|\widetilde{\boldsymbol{K}} - \boldsymbol{K}\|_2 + \zeta\left(\frac{\sqrt{n}}{\mu^2}\|\widetilde{\boldsymbol{K}} - \boldsymbol{K}\|_2\right) \\ &\leq (1 + \zeta)\frac{\sqrt{n}}{\mu^2}\|\widetilde{\boldsymbol{K}} - \boldsymbol{K}\|_2. \end{aligned} \tag{23}$$

We let $\mu_0 = \mu/n$. Substituting the upper bound of $\|\boldsymbol{\alpha}' - \boldsymbol{\alpha}\|_2$ into Equation (22), we can obtain

$$|f'(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \sqrt{n}\kappa(1 + \zeta)\frac{\sqrt{n}}{n^2\mu_0^2}\|\widetilde{\boldsymbol{K}} - \boldsymbol{K}\|_2 = \frac{\kappa(1 + \zeta)}{n\mu_0^2}\|\widetilde{\boldsymbol{K}} - \boldsymbol{K}\|_2. \tag{24}$$

We further introduce a kernel matrix approximation error bound of Nyström method [13] to upper bound $\|\widetilde{\boldsymbol{K}} - \boldsymbol{K}\|_2$.

Theorem 2. *Let $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ be an SPSD matrix. Assume that c columns of $\boldsymbol{K}$ are sampled uniformly at random without replacement, let $\widetilde{\boldsymbol{K}}$ be the rank-k Nyström approximation to $\boldsymbol{K}$, and let $\boldsymbol{K}_k$ be the best rank-k approximation to $\boldsymbol{K}$. For $\epsilon > 0$, $\eta = \sqrt{\frac{\log(2/\delta) g(c,n-c)}{c}}$ with $g(a,s) = \frac{as}{a+s-1/2} \cdot \frac{1}{1-1/(2\max\{a,s\})}$, if $c \geq 64k/\epsilon^4$, then with probability at least $1-\delta$,*

$$\|\boldsymbol{K} - \widetilde{\boldsymbol{K}}\|_F \leq \|\boldsymbol{K} - \boldsymbol{K}_k\|_F + \epsilon \left[\left(\frac{n}{c} \sum_{i \in D(c)} \boldsymbol{K}_{ii} \right) \left(\sqrt{n \sum_{i=1}^{n} \boldsymbol{K}_{ii}^2} + \eta \max(n \boldsymbol{K}_{ii}) \right) \right]^{\frac{1}{2}},$$

where $\sum_{i \in D(c)} \boldsymbol{K}_{ii}$ is the sum of largest c diagonal entries of $\boldsymbol{K}$.

Since $\|\boldsymbol{K} - \widetilde{\boldsymbol{K}}\|_2 \leq \|\boldsymbol{K} - \widetilde{\boldsymbol{K}}\|_F$, if we combine Equation (24) with Theorem 2, we can directly obtain the following theorem.

Theorem 3. *Let $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ be an SPSD matrix. Assume that c columns of $\boldsymbol{K}$ are sampled uniformly at random without replacement, let $\widetilde{\boldsymbol{K}}$ be the rank-k Nyström approximation to $\boldsymbol{K}$, and let $\boldsymbol{K}_k$ be the best rank-k approximation to $\boldsymbol{K}$. For $\epsilon > 0$, $\eta = \sqrt{\frac{\log(2/\delta) g(c,n-c)}{c}}$ with $g(a,s) = \frac{as}{a+s-1/2} \cdot \frac{1}{1-1/(2\max\{a,s\})}$, if $c \geq 64k/\epsilon^4$, then with probability at least $1-\delta$,*

$$|f'(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \frac{\kappa(1+\zeta)}{n\mu_0^2} \left(\|\boldsymbol{K} - \boldsymbol{K}_k\|_F + \epsilon \left[\left(\frac{n}{c} \sum_{i \in D(c)} \boldsymbol{K}_{ii} \right) \left(\sqrt{n \sum_{i=1}^{n} \boldsymbol{K}_{ii}^2} + \eta \max(n \boldsymbol{K}_{ii}) \right) \right]^{\frac{1}{2}} \right),$$

where $\sum_{i \in D(c)} \boldsymbol{K}_{ii}$ is the sum of largest c diagonal entries of $\boldsymbol{K}$.

Theorem 3 measures the effect of kernel matrix approximation on the decision function of LSSVM. It enables us to bound the relative performance of LSSVM when the Nyström method is used to approximate the kernel matrix. We refer to the bound given in Theorem 3 as *a model approximation error bound.*

5 Approximate Model Selection for LSSVM

In order to find the hyperparameters that minimize the generalization error of LSSVM, many model selection approaches have been proposed, such as the cross validation, span bound [17], radius margin bound [5], PRESS criterion [1] and so on. However, when optimizing model selection criteria, all these approaches need to solve LSSVM completely in the inner layer for each iteration.

Here we discuss the problem of approximate model selection. We argue that for model selection purpose, it is sufficient to calculate an approximate criterion that can discriminate the optimal hyperparameters from candidates. Theorem 3 shows that when Nyström approximation is used, the change of learning results of LSSVM is bounded, which is a theoretical support for approximate model selection. In the following, we present an approximate model selection scheme, as shown in Algorithm 2.

We use the RBF kernel $K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \exp\left(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right)$ to describe the scheme, but this scheme is also suitable for other kernel types.

Algorithm 2. Approximate Model Selection Scheme for LSSVM

```
Input: $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$;
Output: $(\gamma, \mu)_{\mathrm{opt}}$;
Initialize: $(\gamma, \mu) = (\gamma^0, \mu^0)$;
repeat
    1: Generate kernel matrix $\boldsymbol{K}$;
    2: Calculate $\boldsymbol{\alpha}$ and $b$ for LSSVM with $\boldsymbol{K}$ and $\mu$ using Algorithm 1;
    3: Calculate model selection criterion $T$ using $\boldsymbol{\alpha}$ and $b$;
    4: Update $(\gamma, \mu)$ to minimize $T$;
until the criterion $T$ is minimized ;
return $(\gamma, \mu)_{\mathrm{opt}}$;
```

Let S denote the iteration steps of optimizing model selection criteria. The complexity of solving LSSVM by calculating the inverse of the exact kernel matrix is $O(n^3)$. For radius margin bound or span bound [5], a standard LSSVM needs to be solved in the inner layer for each iteration, so the total complexity of these two methods is $O(S n^3)$. For PRESS criterion [1], the inverse of kernel matrix also needs to be calculated for each iteration, so its complexity is $O(S n^3)$. From Theorem 1, we know that using Algorithm 1, we could solve LSSVM in $O(c^3 + nck)$. Therefore, if we use the above model selection criteria in the outer layer, the complexity of approximate model selection is $O(S(c^3 + nck))$. For t-fold cross validation, let S_γ and S_μ denote the grid steps of γ and μ. If LSSVM is directly solved, the complexity of t-fold cross validation is $O(tS_\gamma S_\mu n^3)$. However, the complexity of approximate model selection using t-fold cross validation as outer layer criterion will be $O(tS_\gamma S_\mu(c^3 + nck))$.

6 Experiments

In this section, we conduct experiments on several benchmark datasets to demonstrate the effectiveness of approximate model selection.

6.1 Experimental Scheme

The benchmark datasets in our experiments are introduced in [15], as shown in Table 1. For each dataset, there are 100 random training and test pre-defined partitions[1] (except 20 for the Image and Splice dataset). The use of multiple benchmarks means that the evaluation is more robust as the selection of data sets that provide a good match to the inductive bias of a particular classifier becomes less likely. Likewise, the use of multiple partitions provides robustness against sensitivity to the sampling of data to form training and test sets.

In Rätsch's experiment [15], model selection is performed on the first five training sets of each dataset. The median values of the hyperparameters over these five sets are then determined and subsequently used to evaluate the error rates throughout all 100 partitions. However, for this experimental scheme, some of the test data is no longer

[1] http://www.fml.tuebingen.mpg.de/Members/raetsch/benchmark

Table 1. Datasets used in experiments

Dataset	Features	Training	Test	Replications
Thyroid	5	140	75	100
Heart	13	170	100	100
Breast	9	200	77	100
Banana	2	400	4900	100
Ringnorm	20	400	7000	100
Twonorm	20	400	7000	100
Waveform	21	400	4600	100
Diabetes	8	468	300	100
Flare solar	9	666	400	100
German	20	700	300	100
Splice	60	1000	2175	20
Image	18	1300	1010	20

statistically "pure" since it has been used during model selection. Furthermore, the use of median of the hyperparameters would introduce an optimistic bias [3]. In our experiments, we perform model selection on the training set of each partition, then train the classifier with the obtained optimal hyperparameters still on the training set, and finally evaluate the classifier on the corresponding test set. Therefore, we can obtain 100 test error rates for each dataset (except 20 for the Image and Splice dataset). The statistical analysis of these test error rates is conducted to assess the performance of the model selection approach. This experimental scheme is rigorous and can avoid the major flaws of the previous one [3]. All experiments are performed on a Core2 Quad PC, with 2.33GHz CPU and 4GB memory.

6.2 Effectiveness

Following the experimental setup in Section 6.1, we perform model selection respectively using 5-fold cross validation (5-fold CV) and approximate 5-fold CV, that is, approximate model selection by minimizing 5-fold CV error (as shown in Algorithm 2). The CV is performed on a 13×11 grid of (γ, μ) respectively varying in $[2^{-15}, 2^{9}]$ and $[2^{-15}, 2^{5}]$ both with step 2^{2}. We set $c = 0.1n$ and $k = 0.5c$ in Algorithm 1.

We compare effectiveness of two model selection approaches. Effectiveness includes efficiency and generalization. Efficiency is measured by average computation time for model selection. Generalization is measured by the mean test error rate (TER) of the classifiers trained with the optimal hyperparameters produced by different model selection approaches.

Results are shown in Table 2. We use the z statistic of TER [2] to estimate the statistical significance of differences in performance. Let $\bar{x}$ and $\bar{y}$ represent the means of TER of two approaches, and e_x and e_y the corresponding standard errors, then the z statistic is computed as $z = (\bar{x} - \bar{y})/\sqrt{e_x^2 + e_y^2}$ and $z = 1.64$ corresponds to a 95% significance level. From Table 2, approximate 5-fold CV is significantly outperformed by 5-fold CV only on the Splice dataset, but the difference is just 2.5%. Besides, according

Table 2. Comparison of computation time and test error rate (TER) of 5-fold cross validation (5-fold CV) and approximate 5-fold CV

Dataset	5-fold CV		Approximate 5-fold CV	
	Time(s)	TER(%)	Time(s)	TER(%)
Thyroid	1.043	4.680±2.246	0.508	4.800±2.359
Heart	1.127	16.750±3.616	0.623	16.080±3.678
Breast	1.671	27.012±4.636	0.725	26.454±4.675
Banana	7.105	10.758±0.590	1.960	10.941±0.713
Ringnorm	7.601	2.044±0.358	2.058	2.872±3.895
Twonorm	7.097	2.528±0.234	2.213	2.446±0.163
Waveform	7.423	10.172±0.783	2.378	10.352±1.054
Diabetes	10.760	23.583±1.738	2.727	23.406±1.700
Flare solar	19.477	34.230±1.965	5.446	34.230±1.860
German	24.501	23.890±2.231	6.740	23.943±2.304
Splice	42.210	11.326±0.547	14.275	13.862±1.304
Image	141.792	2.876±0.725	28.743	4.628±0.944

to the Wilcoxon signed rank test [7], neither of 5-fold CV and approximate 5-fold CV is statistically superior at the 95% level of significance.

However, Table 2 also shows that approximate 5-fold CV is more efficient than 5-fold CV on all datasets. It is worth noting that the larger the training set size is, the efficiency gain is more obvious, which is in accord with the results of complexity analysis.

7 Conclusion

In this paper, Nyström method was first introduced into the model selection problem. A brand new approximate model selection approach of LSSVM was proposed, which fully exploits the theoretical and computational virtue of Nyström approximation. We designed an efficient algorithm for solving LSSVM and bounded the effect of kernel matrix approximation on the decision function of LSSVM. We derived a model approximation error bound, which is a theoretical support for approximate model selection. We presented an approximate model selection scheme and analyzed its complexity as compared with other classic model selection approaches. This complexity shows the promise of the application of approximate model selection for large scale problems. We finally verified the effectiveness of our approach by rigorous experiments on several benchmark datasets.

The application of our theoretical results and approach to practical large problems will be one of major concerns. Besides, a new efficient model selection criterion directly dependent on kernel matrix approximation will be proposed in near future.

Acknowledgments. The work is supported in part by the Natural Science Foundation of China under grant No. 61170019, and the Natural Science Foundation of Tianjin under grant No. 11JCYBJC00700.

References

1. Cawley, G.C., Talbot, N.L.C.: Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. Neural Networks 17(10), 1467–1475 (2004)
2. Cawley, G.C., Talbot, N.L.C.: Preventing over-fitting during model selection via Bayesian regularisation of the hyper-parameters. Journal of Machine Learning Research 8, 841–861 (2007)
3. Cawley, G.C., Talbot, N.L.C.: On over-fitting in model selection and subsequent selection bias in performance evaluation. Journal of Machine Learning Research 11, 2079–2107 (2010)
4. Chapelle, O., Vapnik, V.: Model selection for support vector machines. In: Advances in Neural Information Processing Systems, vol. 12, pp. 230–236. MIT Press, Cambridge (2000)
5. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Machine Learning 46(1), 131–159 (2002)
6. Cortes, C., Mohri, M., Talwalkar, A.: On the impact of kernel approximation on learning accuracy. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), Sardinia, Italy, pp. 113–120 (2010)
7. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
8. Drineas, P., Mahoney, M.: On the Nyström method for approximating a Gram matrix for improved kernel-based learning. Journal of Machine Learning Research 6, 2153–2175 (2005)
9. Duan, K., Keerthi, S., Poo, A.: Evaluation of simple performance measures for tuning SVM hyperparameters. Neurocomputing 51, 41–59 (2003)
10. Golub, G., Van Loan, C.: Matrix Computations. Johns Hopkins University Press, Baltimore (1996)
11. Guyon, I., Saffari, A., Dror, G., Cawley, G.: Model selection: Beyond the Bayesian / frequentist divide. Journal of Machine Learning Research 11, 61–87 (2010)
12. Higham, N.: Accuracy and stability of numerical algorithms. SIAM, Philadelphia (2002)
13. Kumar, S., Mohri, M., Talwalkar, A.: Sampling techniques for the Nyström method. In: Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS), Clearwater, Florida, USA, pp. 304–311 (2009)
14. Luntz, A., Brailovsky, V.: On estimation of characters obtained in statistical procedure of recognition. Technicheskaya Kibernetica 3 (1969) (in Russian)
15. Rätsch, G., Onoda, T., Müller, K.: Soft margins for AdaBoost. Machine Learning 42(3), 287–320 (2001)
16. Suykens, J., Vandewalle, J.: Least squares support vector machine classifiers. Neural Processing Letters 9(3), 293–300 (1999)
17. Vapnik, V., Chapelle, O.: Bounds on error expectation for support vector machines. Neural Computation 12(9), 2013–2036 (2000)
18. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons, New York (1998)
19. Williams, C., Seeger, M.: Using the Nyström method to speed up kernel machines. In: Advances in Neural Information Processing Systems 13, pp. 682–688. MIT Press, Cambridge (2001)

Exploiting Label Dependency for Hierarchical Multi-label Classification

Noor Alaydie, Chandan K. Reddy, and Farshad Fotouhi

Department of Computer Science
Wayne State University, Detroit, MI, USA
{alaydie,fotouhi}@wayne.edu, reddy@cs.wayne.edu

Abstract. Hierarchical multi-label classification is a variant of traditional classification in which the instances can belong to several labels, that are in turn organized in a hierarchy. Existing hierarchical multi-label classification algorithms ignore possible correlations between the labels. Moreover, most of the current methods predict instance labels in a "flat" fashion without employing the ontological structures among the classes. In this paper, we propose HiBLADE (Hierarchical multi-label Boosting with LAbel DEpendency), a novel algorithm that takes advantage of not only the pre-established hierarchical taxonomy of the classes, but also effectively exploits the hidden correlation among the classes that is not shown through the class hierarchy, thereby improving the quality of the predictions. According to our approach, first, the pre-defined hierarchical taxonomy of the labels is used to decide upon the training set for each classifier. Second, the dependencies of the children for each label in the hierarchy are captured and analyzed using Bayes method and instance-based similarity. Our experimental results on several real-world biomolecular datasets show that the proposed method can improve the performance of hierarchical multi-label classification.

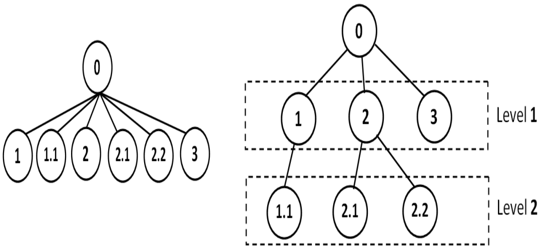
Keywords: Hierarchical multi-label classification, correlation, boosting.

1 Introduction

Traditional classification tasks deal with assigning instances to a single label. In multi-label classification, the task is to find the set of labels that an instance can belong to rather than assigning a single label to a given instance. Hierarchical multi-label classification is a variant of traditional classification where the task is to assign instances to a set of labels where the labels are related through a hierarchical classification scheme [1]. In other words, when an instance is labeled with a certain class, it should also be labeled with all of its superclasses, this is known as the *hierarchy constraint*.

Hierarchical multi-label classification is a widely studied problem in many domains such as functional genomics, text categorization, image annotation and object recognition [2]. In functional genomics (which is the application that we focus on in this paper) the problem is the prediction of gene/protein functions. Biologists have a hierarchical organization of the functions that the genes can be assigned to. An individual gene or protein may be involved in more than one biological activity, and hence, there is a need for a prediction algorithm that is able to identify all the possible functions of a particular

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 294–305, 2012.



(a) Flat (b) Hierarchical

Fig. 1. Flat versus Hierarchical classification. (a) Flat representation of the labels (b) Hierarchical representation of the same set of labels.

gene [2]. There are two types of class hierarchy structures: a rooted tree structure, such as the MIPS's FunCat taxonomy [17], and a directed acyclic graph (DAG) structure, such as the Gene Ontology (GO) [7]. In this paper, we use the FunCat scheme.

Most of the existing research focuses on a "flat" classification approach, that operates on non-hierarchical classification schemes, where a binary classifier is constructed for each label separately as shown in Figure 1(a). This approach ignores the hierarchical structure of the classes shown in Figure 1(b). Reducing a hierarchical multi-label classification problem to a conventional classification problem allows the possibility of applying the existing methods. However, since the prediction of the class labels has to be performed independently, such transformations are not capable of exploiting the interdependencies and correlations between the labels [6]. Moreover, the flat classification algorithm fails to take advantage of the information inherent in the class hierarchy, and hence may be suboptimal in terms of efficiency and effectiveness [9].

1.1 Our Contributions

In this paper, we propose, HiBLADE, a hierarchical multi-label classification framework for modeling the pre-defined hierarchical taxonomy of the labels as well as for exploiting the existing correlations between different labels, that are not given by the taxonomical classification of the labels, to facilitate the learning process.

To the best of our knowledge, *there is no work related to the hierarchical multi-label classification setting that exploits the correlations between different labels other than the domain-based pre-established taxonomical classification of the classes*. Our intuition is that the domain-based taxonomical classification of the classes should be used as additional features while label dependencies are inferred from the data. Specifically, a novel approach to learn the label dependencies using a Bayesian framework and instance-based similarity is proposed. Bayesian framework is used to characterize the dependency relations among the labels represented by the directed acyclic graph (DAG) structure of the Bayesian network. As opposed to other hierarchical multi-label classification algorithms, our algorithm has the following advantages:

1. The underlying pre-defined taxonomy of the labels is explicitly expressed which allows us to gain further insights into the learning problem.

2. It is capable of addressing correlations and interdependencies among the children of a particular label using the Bayesian framework and instance-based similarity.
3. The use of a shared boosting model for the child labels for each label in the hierarchy using the obtained correlations leads to efficient and effective results.

The rest of the paper is organized as follows: related work is discussed in Section 2. Our proposed method, HiBLADE, is presented in Section 3. In Section 4, we report our experimental results on several biomolecular datasets. Then, we conclude and discuss further research directions in Section 5.

2 Related Work

Since conventional classification methods, such as binary classification and multi-class classification, were not designed to directly tackle the hierarchical classification problems, such algorithms are referred to as flat classification algorithms [18]. It is important to mention that flat classification and other similar approaches are not considered to be a hierarchical classification approach, as they create new (meta) classes instead of using pre-established taxonomies.

Different approaches have been proposed in the literature to tackle the hierarchical multi-label classification problem [4,21,16]. Generally, these approaches can be grouped into two categories: the local classifier methods and the global classifier methods. Moreover, most of the existing methods use a top-down class prediction strategy in the testing phase [3,16,18]. The local strategy treats any label independently, and thus ignores any possible correlation or interdependency between the labels. Therefore, some methods perform an additional step to correct inconsistent predictions. For example, in [3], a Bayesian framework is developed for correcting class-membership inconsistency for the separate class-wise models approach. In [1], a hierarchical multi-label boosting algorithm, named HML-Boosting, was proposed to exploit the hierarchical dependencies among the labels. HML-Boosting algorithm relies on the hierarchical information and utilizes the hierarchy to improve the prediction accuracy.

True path rule (TPR) is a rule that governs the annotation of GO and FunCat taxonomies. According to this rule, annotating a gene to a given class is automatically transferred to all of its ancestors to maintain the hierarchy constraint [12]. In [20], a true path ensemble method was proposed. In this method, a classifier is built for each functional class in the training phase. A bottom-up approach is followed in the testing phase to correct the class-membership inconsistency. In a modified version of TPR $(TPR - w)$, a parent weight is introduced. The weight is used to explicitly modulate the contribution of the local predictions with the positive predictions coming from the descendant nodes. In [5], a hierarchical bottom-up Bayesian cost-sensitive ensemble (HBAYES-CS), is proposed. Basically, a calibrated classifier is trained at each node in the taxonomy. H-loss is used in the evaluation phase to predict the labels for a given node. In a recent work [2], we proposed a novel Hierarchical Bayesian iNtegration algorithm HiBiN, a general framework that uses Bayesian reasoning to integrate heterogeneous data sources for accurate gene function prediction. On the other hand, several research groups have studied the effective exploitation of correlation information among different labels in the context of multi-label learning. However, these approaches do not assume the existence of any pre-defined taxonomical structure of the classes [6,11,14,23,24].

3 HiBLADE Algorithm

Let $\mathcal{X} = \Re^d$ be the d-dimensional input space and $\mathcal{Y} = \{y_1, y_2, ..., y_\mathcal{L}\}$ be the finite set of $\mathcal{L}$ possible labels. The hierarchical relationships among classes in $\mathcal{Y}$ are defined as follows: Given $y_1, y_2 \in \mathcal{Y}$, y_1 is the ancestor of y_2, denoted by $(\uparrow y_2) = y_1$, if and only if y_1 is a superclass of y_2.

Let a hierarchical multi-label training set $\mathcal{D} = \{< x_1, \mathcal{Y}_1 >, ..., < x_N, \mathcal{Y}_N >\}$, where $x_i \in \mathcal{X}$ is a feature vector for instance i and $\mathcal{Y}_i \subseteq \mathcal{Y}$ is the set of labels associated with x_i, such that $y_i \in \mathcal{Y}_i \Rightarrow y'_i \in \mathcal{Y}_i, \forall(\uparrow y_i) = y'_i$. Having $\mathcal{Q}$ as the quality criterion for evaluating the model based on the prediction accuracy, the objective function is defined as follows: a function $\mathfrak{f} : \mathcal{D} \rightarrow 2^y$. Here, 2^y is the power set of $\mathcal{Y}$, such that $\mathcal{Q}$ is maximized, and $y' \in \mathfrak{f}(x) \Rightarrow y \in \mathfrak{f}, \forall(\uparrow y') = y$. The function $\mathfrak{f}$ is represented here by the ***HiBLADE*** algorithm.

Hierarchical multi-label learning aims to model and predict $p(child\ class \mid parent\ class)$. Our goal is to make use of hierarchical dependencies as well as the extracted dependencies among the labels y_k where $1 \leq k \leq \mathcal{L}$ and $(\uparrow y_k) = y_m$ such that for each example we can better predict its labels. The problem then becomes how to identify and make use of such dependencies in an efficient way.

3.1 Training Scheme

The training of each classifier is performed locally. During classification, the classifier at each class will only be presented with examples that are positive at the parent class of the current class. Hence, the reached examples are positive examples to the current class and/or to the siblings of that class. In other words, the training for each classifier is performed by feeding as negative training examples, the positive examples at the parent of the current class that are not positive examples at the current class.

3.2 Extending the Features

The feature vector for each example is extended to include the class labels of the levels higher in the hierarchy than the current level as given in line 9 of Algorithm 1. More formally, the feature vector for each instance j that belongs to a class i will have the following form: $f_{i,j} =< x_{j,1}, ..., x_{j,d}, \bar{x}_{j,d+1}, ..., \bar{x}_{j,d+k} >$, where $< x_{j,1}, ..., x_{j,d} >$ is the original feature vector and $< \bar{x}_{j,d+1}, ..., \bar{x}_{j,d+k} >$ are the additional features.

3.3 Label Correlation

The other type of dependencies is modeled using Bayesian structure and instance-based similarity as shown in line 10 of Algorithm 1. For each class i, we get the children classes of class i and $sharedModels$ algorithm (shown in Algorithm 2) is invoked.

Algorithm 1. $HiBLADE$

```
Input: A pair $< \mathcal{Y}, \mathcal{L} >$ where $\mathcal{Y}$ is a tree-structured set of classes and $\mathcal{L}$ is the total number of classes of $\mathcal{Y}$.
Output: For each class $y_l \in \mathcal{Y}$, the final composite classifier $y_l = sign[F_l(x)]$.
Algorithm:
for $i = 1, ..., \mathcal{L}$ do
    if class $i$ is a leaf class then
        Do nothing
    else
        Let $children(i) = y_{1i}, ..., y_{ki}$ be the $k$ children classes of $i$
        Form the new feature vectors by adding the labels of the classes at the higher levels to the current feature vectors.
        Learn classifiers for $children(i)$ using the shared models Algorithm
    end if
end for
```

In each boosting iteration t, the entire pool is searched for the best fitted model other than the model that was built directly for that label and its corresponding combination weights, the best fitted model is called h_l^t. We refer to the best fitted model as the candidate model. The chosen model h_c^t is then updated based on the following formula:

$$\gamma_{ij} = \frac{\epsilon_{ii}}{\epsilon_{ii} + \epsilon_{ji}} * \beta_{ij} \tag{1}$$

where ϵ_{ji} is the error results from applying model h_j^t on the examples in class i and ϵ_{ii} is the error results from applying the model h_i^t on the examples in class i. β_{ij} controls the proportional contribution of Bayesian-based and instance-based similarities. β_{ij} is computed as follows:

$$\beta_{ij} = \phi * b_{ij} + (1 - \phi) * s_{ij} \tag{2}$$

where b_{ij} is the Bayesian correlation between class i and class j, and it is estimated as $b_{ij} = |i \cap j| / |j|$, where $|i \cap j|$ is the number of positive examples in class i and class j and $|j|$ is the number of positive examples in class j. s_{ij} is the instance-based similarity between class i and class j. Each instance from one class is compared to each other instance from the other class. In HiBLADE, s_{ij} is computed using the Euclidean distance between the positive examples in both classes that has the following formula:

$$s_{ij} = \sqrt{\sum_l (i_l - j_l)^2} \tag{3}$$

where l is the corresponding feature in the two vectors. s_{ij} is normalized to be in the range of $[0, 1]$. ϕ is a threshold parameter that has a value in the range $[0, 1]$. Setting ϕ to 0 means that only instance-based similarity is taken into consideration in the learning process. While setting it to 1 means that only Bayesian-based correlation is taken into consideration. On the other hand, any value of ϕ between 0 and 1 combines both types of correlation. It is important to emphasize that these computations are performed only for the class that is found to be the most useful class with respect to the current class.

In the general case, both classes, the current class and the candidate class, contribute to the final prediction. In other words, any value of ϵ_{ji} other than 0, indicates the level of contribution from the candidate class. More specifically, if the error of the candidate class, ϵ_{ji}, is greater than the error of the current class, ϵ_{ii}, the value of γ_{ij} will be small indicating that only a limited contribution of the candidate class is considered. In contrast, if the error of the current class, ϵ_{ii}, is greater than the error of the candidate class, ϵ_{ji}, then γ_{ij} will be high, and hence, the prediction decision will be dependable more on the candidate class. Finally, the models for the current class and the used candidate class are replaced by the new learned models. At the end, the composite classifiers F_c provide the prediction results.

Algorithm 2 shows the details of the shared models algorithm. The shared models algorithm takes as input the children classes of a particular class together with the feature vectors for the instances that are positive at the parent class. These instances will form the positive and negative examples for each one of the children classes. The algorithm begins by initializing a pool of M models, where M is the number of children classes, one for each class that is learned using a boosting-type algorithm such as ADABOOST. The number of base models to be generated is determined by T. In each iteration t and for each label in the set of the children labels, we look for the best fitted model, $h_l^t(x)$ and the corresponding combination weights, α_l^t. The contribution of the selected base model, $h_l^t(x)$, to the overall classifier, $F_c(x)$, depends on the current label. In other words, if the error, ϵ_{ji} of the candidate classifier is 0, this will be a perfect model for the current label. Hence, equation (1) will be reduced to $\gamma_{ij} = \beta_{ij}$. In this case, the contribution of that model depends on the level of correlation between the candidate class and the current one. On the other hand, if the current model is a perfect model, i.e., the error $\epsilon_{ii} = 0$, then equation 1 will be reduced to $\gamma_{ij} = 0$, which means that for the current iteration, there is no need to look at any other classifier.

Algorithm 2. *SharedModels*

```
Input: $D = \{(x_i, Y_i) : i = 1, ..., N\}$, where $x_i \in X$ is a feature vector for instance $i$ and
  $Y_i \subset Y$ is the set of labels associated with $x_i$ and $M$ is the number of labels under study. $\phi$:
  a threshold parameter.
Output: $y_c = sign[F_c(x)]$
Algorithm:
Set $F_c(x) = 0$ for each label $c = 1, .., M$
Initialize a pool of candidate shared models: $SM_p = h_1(.), ..., h_M(.)$ where $h_i(.)$ is a model
  learned on the label $i$ using the boosting-type algorithm.
for $t = 1, ..., T$ do
    for $c = 1, ..., M$ do
        Find $\alpha_l^t$ and $h_l^t \in SM_p$ where $c \neq l$ that minimize the loss function on label $c$.
        $F_c(x) = F_c(x) + h_c^t(x) * \alpha_c^t * (1 - \gamma_{cl}) + h_l^t(x) * \alpha_l^t * \gamma_{cl}$
        Replace $h_c^t(x)$ and $h_l^t(x)$ in $SM_p$ with the new learned models using the boosting-type
          algorithm.
    end for
end for
```

4 Experimental Details

We chose to demonstrate the performance of our algorithm for the prediction of gene functions in yeast using four bio-molecular datasets that were used in [20]. Valentini [20] pre-processed the datasets so that for each dataset, only genes that are annotated with FunCat taxonomy are selected. To make this paper self-contained, we briefly explain the data collection process and the pre-processing steps performed on the data. Uninformed features that have the same value for all of the examples are removed. Class "99" in FunCat corresponds to an "unclassified protein". Therefore, genes that are annotated only with that class are excluded. Finally, in order to have a good size of positive training examples for each class, selection has been performed to classes with at least 20 positive examples. Dataset characteristics are summarized in Table 1.

Table 1. The characteristics of the four bio-molecular datasets used in our experiments

Dataset	Description	Samples	Features	classes
Gene-Expr	Gene expression data	4532	250	230
PPI-BG	PPI data from BioGRID	4531	5367	232
Pfam-1	Protein domain binary data	3529	4950	211
PPI-VM	PPI data from Von Mering experiments	2338	2559	177

The gene expression dataset, Gene-Expr, is obtained by merging the results of two studies, gene expression measures relative to 77 conditions and transcriptional responses of yeast to environmental stress measured on 173 conditions [10]. For each gene product in the protein-protein interaction dataset, PPI-BG, a binary vector is generated that implies the presence or absence of protein-protein interaction. Protein-protein interaction data have been downloaded from BioGRID database [19,20]. In Pfam-1 dataset, a binary vector is generated for every gene product that reflects the presence or absence of 4950 protein domains obtained from Pfam (Protein families) database [8,20]. For PPI-VM dataset, Von Mering experiments produced protein-protein data from yeast two-hybrid assay, mass spectrometry of purified complexes, correlated mRNA expression and genetic interactions [22].

Table 2. Per-level F_1 measure for Gene-Expr dataset using Flat, $HiBLADE_I$, $HiBLADE_C$ with ϕ = 0.5 and $HiBLADE_B$ for boosting iterations=50

Level	Flat	$HiBLADE_I$	$HiBLADE_C$	$HiBLADE_B$
		$\phi = 0.0$	$\phi = 0.5$	$\phi = 1.0$
1	**0.3537**	0.2328	0.2301	0.2336
2	0.1980	0.4052	**0.4427**	0.4094
3	0.1000	0.3575	**0.4019**	0.3742
4	0.2000	0.2714	**0.3598**	0.2874

4.1 Evaluation Metrics

Classical evaluation measures such as precision, recall and F-measure are used by unstructured classification problems and thus, they are inadequate to address the hierarchical natures of the classes. Another approach that is used for the hierarchical multi-label learning is to use extended versions of the single label metrics (precision, recall and F-measure). To evaluate our algorithm, we adopted both, the classical and the hierarchical evaluation measures. F_1 measure considers the joint contribution of both precision (P) and recall (R). F_1 measure is defined as follows:

$$F_1 = \frac{2 \times P \times R}{P + R} = \frac{2TP}{2TP + FP + FN} \tag{4}$$

where TP stands for True Positive, TN for True Negative, FP for False Positive and FN for False Negative. When TP=FP=FN=0, we made F_1 measure to equal to 1 as the classifier has correctly classified all the examples as negative examples [9]. Hierarchical measures are defined as follows:

$$hP = \frac{1}{|l(P(x))|} \sum_{p \in l(P(x))} \frac{|C(x) \cap \uparrow p|}{|\uparrow p|} \tag{5}$$

$$hR = \frac{1}{|l(C(x))|} \sum_{c \in l(C(x))} \frac{|\uparrow c \cap P(x)|}{|\uparrow c|} \tag{6}$$

$$hF = \frac{2 \times hP \times hR}{hP + hR} \tag{7}$$

where hP, hR and hF stands for hierarchical precision, hierarchical recall and hierarchical F-measure, respectively. $P(x)$ is a subgraph formed by the predicted class labels for the instance x while $C(x)$ is a subgraph formed by the true class labels for the instance x. p is one of the predicted class labels and c is one of the true labels for instance x. $l(P(x))$ and $l(C(x))$ are the set of leaves in graphs $P(x)$ and $C(x)$, respectively. We also computed both micro-averaged hierarchical F-measure (hF_1^{μ}) and macro-averaged hierarchical F-measure hF_1^M. hF_1^{μ} is computed by computing hP and hR for each path in the hierarchical structure of the tree and then applying equation (7). On the other hand, hF_1^M is computed by calculating hF_1 for each path in the hierarchical structure of the classes independently and then averaging them. Having high hierarchical precision means that the predictor is capable of predicting the most general functions of the instance, while having high hierarchical recall indicates that the predictor is able to predict the most specific classes [20]. The hierarchical F-measure takes into account the partially correct paths in the overall taxonomy.

4.2 Experimental Results and Discussion

We analyzed the performance of the proposed framework at each level of the FunCat taxonomy, and we also compared the proposed method with four other methods that follow the local classifier approach, namely, HBAYES-CS, HTD, TPR and TPR-w.

Table 3. Per-level F_1 measure for PPI-BG dataset using Flat, $HiBLADE_I$, $HiBLADE_C$ with $\phi = 0.5$ and $HiBLADE_B$ for boosting iterations=50

Level	Flat	$HiBLADE_I$	$HiBLADE_C$	$HiBLADE_B$
		$\phi = 0.0$	$\phi = 0.5$	$\phi = 1.0$
1	0.0808	0.2014	0.1833	**0.2052**
2	0.0267	0.6904	0.6984	**0.6998**
3	0.0001	0.6446	0.6304	**0.6520**
4	0.0001	0.6743	0.6454	**0.6747**

HBAYES-CS, TPR and TPR-w are described in the Related Work Section. HTD (Hierarchical Top-Down) is the baseline method that belongs to the local classifier strategy and performs hierarchical classification in a top-down fashion. Since HiBLADE also belongs to the local classifier strategy, it is fair to have a comparison against a local classifier approach that does not consider any type of correlation between the labels. We also analyzed the effect of the proper choice of the threshold ϕ on the performance of the algorithm. The setup for the experiments is summarized as follows:

- Flat: This is the baseline method that does not take the hierarchical taxonomy of the classes into account and does not consider label dependencies. A classifier is built for each class independently of the others. We used AdaBoost as the base learner to form a baseline algorithm for the comparison with the other methods.
- $HiBLADE_I$: The proposed algorithm that considers Instance-based similarities only. Here ϕ is set to zero.
- $HiBLADE_B$: The proposed algorithm that considers classes correlation based on Bayesian probabilities only. Here ϕ is set to one.
- $HiBLADE_C$: The proposed algorithm that considers a combination of both instance-based similarity and classes correlation. Here ϕ is set to 0.5.

Table 4. Per-level F_1 measure for $Pfam - 1$ dataset using Flat, $HiBLADE_I$, $HiBLADE_C$ with $\phi = 0.5$ and $HiBLADE_B$ for boosting iterations=50

Level	Flat	$HiBLADE_I$	$HiBLADE_C$	$HiBLADE_B$
		$\phi = 0.0$	$\phi = 0.5$	$\phi = 1.0$
1	**0.1133**	0.0924	0.0827	0.1085
2	0.0267	0.8524	**0.8702**	0.7273
3	0.1000	0.7473	**0.7946**	0.6824
4	0.2222	0.5122	**0.5135**	0.5085

First, we performed a level-wise analysis of the F-measure of the FunCat classification tree on the four datasets. In measuring the level-wise performance, level 1 reflects the root nodes while all other classes are at depth i, where $2 \leq i \leq 5$. We show the results for the top four levels in the hierarchy for the proposed method and the flat method. Moreover, we show the performance of the proposed framework with different ϕ's values while setting the number of boosting iterations to 50 iterations. Tables 2, 3, 4 and 5 show the results of per-level evaluation for Gene-Expr, PPI-BG, Pfam-1 and PPI-VM datasets, respectively. The most significant measures for each level are highlighted.

The proposed algorithm outperforms the flat classification method in most of the cases with significant differences in the performance measurements. The results in Tables 2, 3, 4 and 5 indicate that the deeper the level the better the performance of the proposed algorithm compared to the flat classification method. For example, in all of the datasets, the proposed algorithm outperformed the flat classification method in all the levels that are higher than level 1. This result is consistent with our understanding of both of the classification schemes. In other words, the proposed method and the flat classification method have a similar learning procedure for the classes in the first level. However, the proposed method achieved better results for the deeper levels in the hierarchy.

Table 5. Per-level F_1 measure for PPI-VM dataset using Flat, $HiBLADE_I$, $HiBLADE_C$ with $\phi = 0.5$ and $HiBLADE_B$ for boosting iterations=50

Level	Flat	$HiBLADE_I$ $\phi = 0.0$	$HiBLADE_C$ $\phi = 0.5$	$HiBLADE_B$ $\phi = 1.0$
1	**0.1631**	0.1266	0.1029	0.1193
2	0.1786	0.6033	**0.6758**	0.6601
3	0.0001	0.5802	0.6822	**0.6957**
4	0.0001	**0.6931**	0.5246	0.5417

To get more insights into the best choice of ϕ threshold, we compare hierarchical precision, hierarchical recall, hierarchical F_1^μ measure and hierarchical F_1^M measure for Gene-Expr, PPI-BG, Pfam-1 and PPI-VM datasets for $\phi = 0.0, 0.5$ and 1.0, respectively, for 50 boosting iterations. Table 6 shows the results of the comparisons. The most significant measures are highlighted. As shown in Table 6, the combination of Bayesian-based correlation and instance-based similarity achieved the best performance results in most of the cases. For example, six of the highest performance values, in general, in this table are achieved when $\phi = 0.5$.

Table 6. Hierarchical precision, hierarchical recall, hierarchical F_1^M and hierarchical F_1^μ measures of HiBLADE for all the four datasets using boosting iterations =50

Measure	Gene-Expr			PPI-BG		
	$\phi = 0.0$	$\phi = 0.5$	$\phi = 1.0$	$\phi = 0.0$	$\phi = 0.5$	$\phi = 1.0$
hP	0.820	0.808	**0.826**	0.878	**0.924**	0.875
hR	**0.644**	0.630	0.627	0.662	0.686	**0.701**
hF_1^M	**0.702**	0.689	0.692	0.735	**0.769**	0.756
hF_1^μ	**0.722**	0.708	0.712	0.755	**0.787**	0.778

Measure	Pfam-1			PPI-VM		
	$\phi = 0.0$	$\phi = 0.5$	$\phi = 1.0$	$\phi = 0.0$	$\phi = 0.5$	$\phi = 1.0$
hP	0.763	0.836	**0.875**	0.716	**0.748**	0.719
hR	0.625	**0.663**	0.637	0.542	0.551	**0.557**
hF_1^M	0.669	**0.720**	0.714	0.590	**0.605**	0.601
hF_1^μ	0.687	**0.740**	0.737	0.617	**0.635**	0.628

Furthermore, we conducted comparisons of hierarchical F-measure with HBAYES-CS, HTD, TPR and TPR-w methods. HBAYES-CS is using Guassian SVMs as the base learners, while HTD, TPR and TPR-w are using Linear SVMs as the base learners. Figure 2 shows the F-measure of the different methods. By exploiting the label dependencies, the classifiers performance are effected positively. Our results show that the proposed algorithm significantly outperforms the local learning algorithms. Although there is no clear winner among the different versions of HiBLADE algorithm, HiBLADE always achieved significantly better results than the other methods.

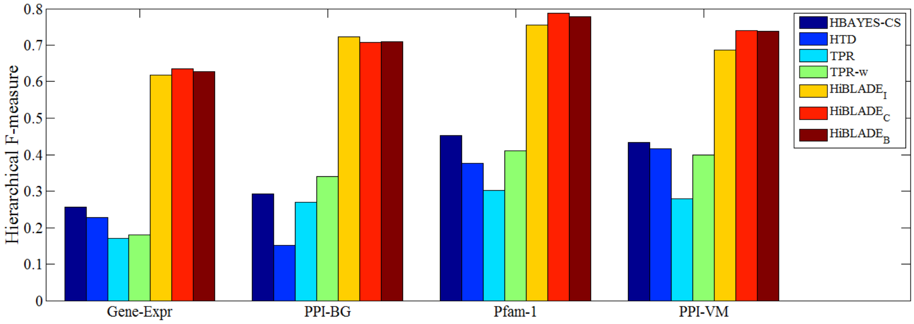


Fig. 2. Hierarchical F-measure comparison between HBAYES-CS, HTD, TPR, TPR-w, $HiBLADE_I$, $HiBLADE_C$ and $HiBLADE_B$. For the $HiBLADE$ algorithm, the number of boosting iterations is 50 and $\phi = 0.5$ for $HiBLADE_C$.

5 Conclusion

In this paper, we proposed a hierarchical multi-label classification framework for incorporating information about the hierarchical relationships among the labels as well as the label correlations. The experimental results showed that the proposed algorithm, HiBLADE, outperforms the flat classification method and the local classifiers method that builds independent classifier for each class. For future work, we plan to generalize the proposed approach to general graph structures and develop more scalable solutions using some other recent proposed boosting strategies [13,15].

References

1. Alaydie, N., Reddy, C.K., Fotouhi, F.: Hierarchical boosting for gene function prediction. In: Proceedings of the 9th International Conference on Computational Systems Bioinformatics (CSB), Stanford, CA, USA, pp. 14–25 (August 2010)
2. Alaydie, N., Reddy, C.K., Fotouhi, F.: A Bayesian Integration Model of Heterogeneous Data Sources for Improved Gene Functional Inference. In: Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine (ACM-BCB), Chicago, IL, USA, pp. 376–380 (August 2011)
3. Barutcuoglu, Z., Schapire, R.E., Troyanskaya, O.G.: Hierarchical multi-label prediction of gene function. Bioinformatics 22(7), 830–836 (2006)
4. 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)

5. Cesa-Bianchi, N., Valentini, G.: Hierarchical cost-sensitive algorithms for genome-wide gene function prediction. In: Proceedings of the Third International Workshop on Machine Learning in Systems Biology, Ljubljana, Slovenia, pp. 25–34 (2009)
6. Cheng, W., Hüllermeier, E.: Combining instance-based learning and logistic regression for multilabel classification. Machine Learning 76(2-3), 211–225 (2009)
7. The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. Nature Genetics 25(1), 25–29 (2000)
8. Deng, M., Chen, T., Sun, F.: An integrated probabilistic model for functional prediction of proteins. In: Proc. 7th Int. Conf. Comp. Mol. Biol., pp. 95–103 (2003)
9. Esuli, A., Fagni, T., Sebastiani, F.: Boosting multi-label hierarchical text categorization. Information Retrieval 11, 287–313 (2008)
10. Gasch, A.P., Spellman, P.T., Kao, C.M., Carmel-Harel, O., Eisen, M.B., Storz, G., Botstein, D., Brown, P.O.: Genomic expression programs in the response of yeast cells to environmental changes. Mol. Biol. Cell 11, 4241–4257 (2000)
11. Jun, G., Ghosh, J.: Multi-class Boosting with Class Hierarchies. In: Benediktsson, J.A., Kittler, J., Roli, F. (eds.) MCS 2009. LNCS, vol. 5519, pp. 32–41. Springer, Heidelberg (2009)
12. Mostafavi, S., Morris, Q.: Using the gene ontology hierarchy when predicting gene function. In: Conference on Uncertainty in Artificial Intelligence (UAI), Montreal, Canada, pp. 22–26 (September 2009)
13. Palit, I., Reddy, C.K.: Scalable and Parallel Boosting with MapReduce. IEEE Transactions on Knowledge and Data Engineering, TKDE (in press, 2012)
14. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier Chains for Multi-label Classification. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009. LNCS, vol. 5782, pp. 254–269. Springer, Heidelberg (2009)
15. Reddy, C.K., Park, J.-H.: Multi-resolution Boosting for Classification and Regression Problems. Knowledge and Information Systems (KAIS) 29(2), 435–456 (2011)
16. Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylor, J.: Kernel-Based Learning of Hierarchical Multilabel Classification Models. The Journal of Machine Learning Research 7, 1601–1626 (2006)
17. Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., Tetko, I., Güldener, U., Mannhaupt, G., Münsterkötter, M., Mewes, H.W.: The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. Nucleic Acids Research 32(18), 5539–5545 (2004)
18. Silla Jr., C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery 22, 31–72 (2011)
19. Stark, C., Breitkreutz, B., Reguly, T., Boucher, L., Breitkreutz, A., Tyers, M.: BioGRID: a general repository for interaction datasets. Nucleic Acids Research 34, D535–D539 (2006)
20. Valentini, G.: True path rule hierarchical ensembles for genome-wide gene function prediction. IEEE ACM Transactions on Computational Biology and Bioinformatics 8(3), 832–847 (2011)
21. Vens, C., Struyf, J., Schietgat, L., Džeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Machine Learning 73, 185–214 (2008)
22. Von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S., Fields, S., Bork, P.: Comparative assessment of large-scale data sets of protein-protein interactions. Nature 417, 399–403 (2002)
23. Yan, R., Tesic, J., Smith, J.R.: Model-Shared Subspace Boosting for Multi-label Classification. In: 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), New York, NY, USA, pp. 834–843 (2007)
24. Zhang, M.-L., Zhang, K.: Multi-label learning by exploiting label dependency. In: Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2010), Washington, D.C., USA, pp. 999–1007 (2010)

Diversity Analysis on Boosting Nominal Concepts

Nida Meddouri[1], Héla Khoufi[1], and Mondher Sadok Maddouri[2]

[1] Research Unit on Programming, Algorithmics and Heuristics - URPAH,
Faculty of Science of Tunis - FST,
Tunis - El Manar University
{nida.meddouri,hela.khoufi}@gmail.com

[2] College of Community, Hinakiyah
Taibah University - Medinah Monawara
Kingdom of Saoudi Arabia
maddourimondher@yahoo.fr

Abstract. In this paper, we investigate how the diversity of *nominal classifier* ensembles affects the *AdaBoost* performance [13]. Using 5 real data sets from the *UCI Machine Learning Repository* and 3 different diversity measures, we show that *Q Statistic* measure is mostly correlated with *AdaBoost* performance for 2-class problems. The experimental results suggest that the performance of *AdaBoost* depend on the *nominal classifier* diversity that can be used as a stopping criteria in ensemble learning.

1 Introduction

Boosting is an adaptive approach, which makes it possible to correctly classify an object that can be badly classified by an ordinary classifier. The main idea of *Boosting* is to build many classifiers who complement each other, in order to build a more powerful classifier. *Adaboost* (***Ada****ptive* ***Boost****ing*) is the most known method of *Boosting* for classifiers generation and combination.

AdaBoost algorithm is iterative. At first, it selects a subset of instances from the learning data set (different subset from the training data set in each iteration). Then, it builds a classifier using the selected instances. Next, it evaluates the classifier on the learning data set, and it starts again T times.

It has been found that this ingenious manipulation of training data can favorise diversity especially for *linear* classifiers [11]. However, there is no study concerning the role of diversity on *Nominal Concepts* classifiers [13]. In this paper, we study how diversity changes according to the *nominal classifier* numbers and we show when adding new classifiers to the team can't provide further improvements.

This paper is organized as follows: section 2 presents the principle of *Classifier of Nominal Concepts* (***CNC***) used in *Boosting* [7,13]. In section 3, we discuss the diversity of classifiers and the different measures that can be exploited in the classifiers ensembles generation. Section 4 presents the experimental results that prove when the diversity can be useful in *Boosting of Nominal Concepts*.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 306–317, 2012.

2 Boosting of CNC

CNC is a classifier based on the *Formal Concept Analysis*. It is distinguished from the other *Formal Concept Analysis* methods by handling nominal data. It generates *Nominal Concept* that is used as classification rule. Comparative studies and experimental results have proved the benefits of *CNC* compared to existing ones (*GRAND, RULEARNER, CITREC, IPR*) [13].

2.1 Nominal Concepts

A nominal classifier can be build using the whole of training instances $\mathcal{O} = \{o_1, ..., o_N\}$ described by L nominal attributes $\mathcal{AN}$ (which are not necessary binary).

$$AN = \{AN_l | l = \{1, .., L\}\}. \tag{1}$$

At first, the pertinent nominal concept AN^* is extracted from the training instances by selecting the nominal attribute which minimises the measure of *Informational Gain* [13]. Then, the associated instances are selected with each value v_j (j = {1,..,J} and J the number of different value of a nominal attribut) from this attribute as $\delta(AN^* = v_j)$. The δ operator is defined by:

$$\delta(AN^* = v_j) = \{o \in O | AN^*(o) = v_j\}. \tag{2}$$

Then, the other attributes describing all the extracted instances are determined (using the closure operator $\delta \circ \varphi$ $(AN^* = v_j)$) as follows:

$$\varphi(B) = \{v_j | \forall o, o \in B \text{ } and \text{ } \exists AN_l \in AN | AN_l(o) = v_j\}. \tag{3}$$

In [13], a method called *BNC* (***B**oosting **N**ominal **C**oncepts*) has been proposed. The advantage of *BNC* is to build a part of the lattice covering *the best nominal concepts* (*the pertinent*) which is used as classification rules (the *Classifier Nominal Concepts*). The *BNC* has the particularity to decide the number of *nominal classifiers* in order to control the time of application and to provide the best decision.

2.2 Learning Concept Based Classifiers

For K-class problem, let $Y = \{1, .., K\}$ the class labels, with $y_i \in Y$ is the class label associated for each instance o_i (*i=1 to N*). To generate T classifiers in *AdaBoost*, the distribution of the weight of o_i is initially determined as :

$$D_0(i) = (1/N). \tag{4}$$

The weight of o_i is:

$$w^1_{i,y} = D_0(i)/(K-1) \text{ } for \text{ } each \text{ } y \in Y - \{y_i\} \tag{5}$$

On each iteration t from 1 to T, we define:

$$W_i^t = \sum_{y \neq y_i} w_{i,y}^t \quad and \quad we \quad set \quad q_t(i,y) = \frac{w_{i,y}^t}{W_i^t} \ for \ each \ y \neq y_i \tag{6}$$

The distribution of weights is calculated by:

$$D_t(i) = \frac{W_i^t}{\sum_{i=1}^N W_i^t} \tag{7}$$

Each generated nominal classifier h_t provides an estimated probability $p_t(o_i,y_i)$ to the class y_i from the entry o_i. Three cases are presented:

- If $p_t(o_i,y_i) = 1$ and $p_t(o_i,y) = 0$, $\forall y \neq y_i$, h_t has correctly predicted the class of o_i.
- If $p_t(o_i,y_i) = 0$ and $p_t(o_i,y) = 1$, $\forall y \neq y_i$, h_t has an opposed prediction of the class of o_i.
- If $p_t(o_i,y_i) = p_t(o_i,y)$, $\forall y \neq y_i$, the class of o_i is selected randomly (y or y_i).

The error rate of h_t is calculated on the weighted training set. If an instance o_i is correctly classified by h_t, then the weight of this instance is reduced. Otherwise, the weightis increased. The pseudo-loss of the classifier h_t is defined as:

$$\epsilon_t = 0.5 \times \sum_{i=1}^N D_t(i)(1 - p_t(o_i, y_i) + \sum_{y \neq y_i} q_t(i, y) p_t(o_i, y)) \tag{8}$$

The weights are then updated according to β_t:

$$\beta_t = \varepsilon_t / (1 - \varepsilon_t) \tag{9}$$

The procedure is repeated T times and the final result of *BNC* is determined via the combination of the generated classifier outputs:

$$h_{fin}(o_i) = \arg\max_{y \in \mathcal{Y}} \sum_{t=1}^T log(1/\beta_t) \times p_t(o_i, y_i). \tag{10}$$

The first variant of the *AdaBoost* algorithm is called *Adaboost.M1* [5,6] that uses the previous process and stops it when the error rate of a classifier becomes over 0.5. The second variant is called *AdaBoost.M2* [6] which has the particularity of handling multi-class data and operating whatever the error rate is. In this study, we use *AdaBoost.M2* since *Adaboost.M1* has the limit to stop *Boosting* if the learning error exceeds 0.5. In some experiments, Adaboost.M1 can be stopped after the generation of first classifier thus we cannot calculate the diversity of classifier ensemble in this particular case.

Recent researches have proved the importance of classifier diversity in improving the performance of *AdaBoost* [1,4,8]. We shall discuss about that in the next section.

3 Classifier Diversity

According to [4], *linear classifiers* should be different from each other, otherwise the decision of the ensemble will be of lower quality than the individual decision. This difference, also called *diversity*, can lead to better or worse overall decision [3].

In [14], the authors found a consistent pattern of diversity showing that at the beginning, the generated *linear classifiers* are highly diverse but as the learning progresses, the diversity gradually returns to its starting level. This suggests that it could be beneficial to stop *AdaBoost* before diversity drops. The authors confirm that there are a consistent patterns of diversity with many measures using *linear classifiers*. However, they report that the pattern might change if other classifier models are used. In the paper, we will prove that this pattern is the same with *nominal classifiers*.

Many measures can be used to determine the diversity between classifiers [11]. In this section, we present three(3) of them: $\mathcal{Q}$ *Statistic*, *Correlation Coefficient* ($\mathcal{CC}$) and *Pairwise Interrater Agreement* (*kp*). These pairwise measures have the same diversity value (0) when the classifiers are statistically independent. They are called pairwise because they consider the output classifiers, two at a time and then they average the calculated pairwise diversity. These measures are computed based on the agreement and the disagreement between each 2 classifiers (see Table 1).

Table 1. The agreement and disagreement between two classifiers

	h$_k$ correct (1)	h$_k$ incorrect (0)
h$_j$ correct (1)	N^{11}	N^{10}
h$_j$ incorrect (0)	N^{01}	N^{00}

$$N{=}N^{00}{+}N^{01}{+}N^{10}{+}N^{11}$$

N^{vw}(v,w=1,0) is the number of instances correctly or incorrectly classified by the two classifiers: h$_j$ and h$_k$ (j,k = $1..T$).

The Q Statistic: Using table 1, this measure is calculated as follows:

$$\mathcal{Q}_{j,k} = \frac{N^{11}N^{00} - N^{10}N^{01}}{N^{11}N^{00} + N^{10}N^{01}} \tag{11}$$

$\mathcal{Q}$ *Statistic* varies between -1 and 1. Classifiers that tend to recognize correctly the same instances, have positive $\mathcal{Q}$ values, and classifiers that commit errors on different instances have negative $\mathcal{Q}$ values. In [11], the authors showed that the negative dependency of linear classifiers can offer a dramatic improvement in *Boosting*.

The Correlation Coefficient: The correlation between 2 classifiers is given by:

$$\rho_{j,k} = \frac{N^{11}N^{00} - N^{10}N^{01}}{\sqrt{(N^{11}+N^{10})(N^{01}+N^{00})(N^{11}+N^{01})(N^{10}+N^{00})}} \tag{12}$$

The Pairwise Interrater agreement: For this measure, the agreement between each pair of classifiers is calculated as:

$$kp_{j,k} = 2\frac{N^{11}N^{00} - N^{10}N^{01}}{(N^{11}+N^{10})(N^{01}+N^{00})(N^{11}+N^{01})(N^{10}+N^{00})} \tag{13}$$

For any pair of classifiers, $\mathcal{Q}$ and ρ have the same sign. The maximum value of ρ and kp is 1 but the minimum value depends on the individual performance of the classifiers.

In [11], it is reported that there is not unique choice of diversity measure. But, for linear classifiers, the authors recommended the use of $\mathcal{Q}$ *Statistic* for it's simplicity and it's significant results. Then, it's interesting to compare the previous measures in *Boosting Nominal Concept*.

4 Experimental Study

The goal of this section is to study the relationship between the *nominal classifiers* diversity and *AdaBoost* performance for 2-class problems.

Table 2. Characteristics of used data sets

Data Sets	*Instances*	*Attributes*	*Data diversity*
Credit German	1000	20	**98.59**%
Diabetes	768	8	22.83%
Ionosphere	351	4	**90.2**%
Tic Tac Toe	958	9	**100**%
Transfusion	748	4	1.07%

The experiments are performed on 5 real data sets extracted from *UCI Machine Learning Repository*[1] [2] and the algorithms are implemented in `WEKA`[2], a widely used toolkit.

The characteristics of these data sets are reported in Table 2. For each data set, we respectively give the number of instances and the number of attributes. Also, we present the *data diversity* rate that indicates the samples which are different (including the class label) in the data [9].

[1] `http://archive.ics.uci.edu/ml/`

[2] `http://www.cs.waikato.ac.nz/ml/weka/`

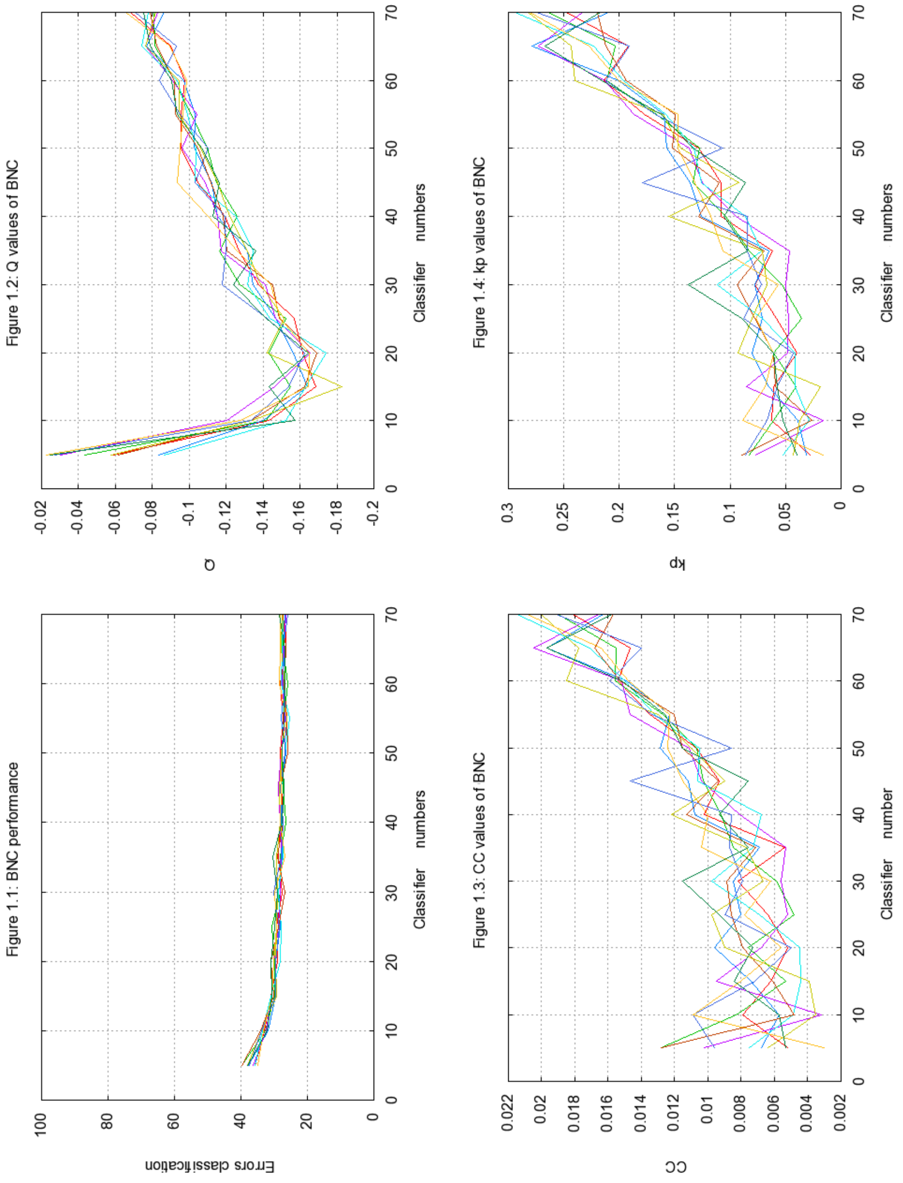


Fig. 1. Diversity and error rates of BNC on Credit German

The performance of *BNC* is evaluated in terms of error rates. To calculate this performance, we report the average of 10 experimentations. Each experiment was performed using *10 cross-validations*, that is the most used method in the literature for validation [10].

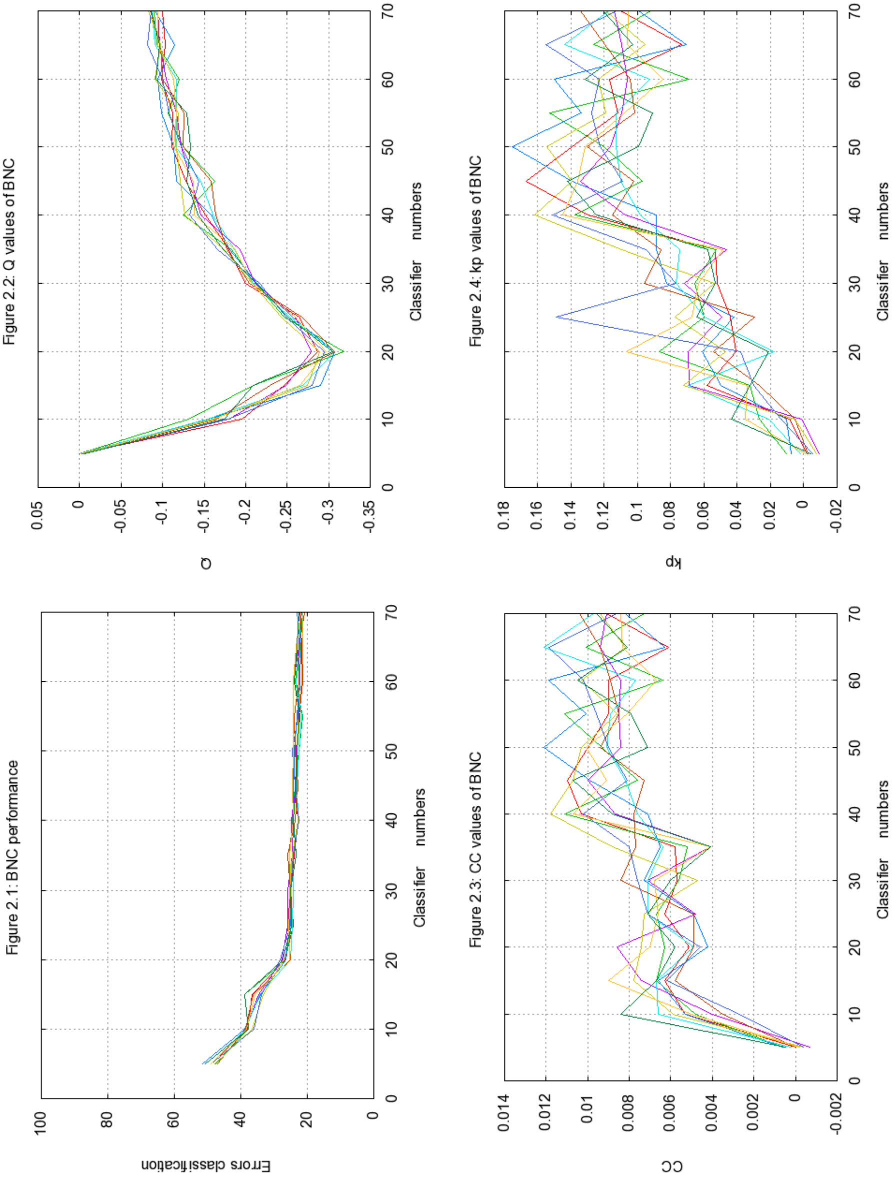


Fig. 2. Diversity and error rates of BNC on Diabetes

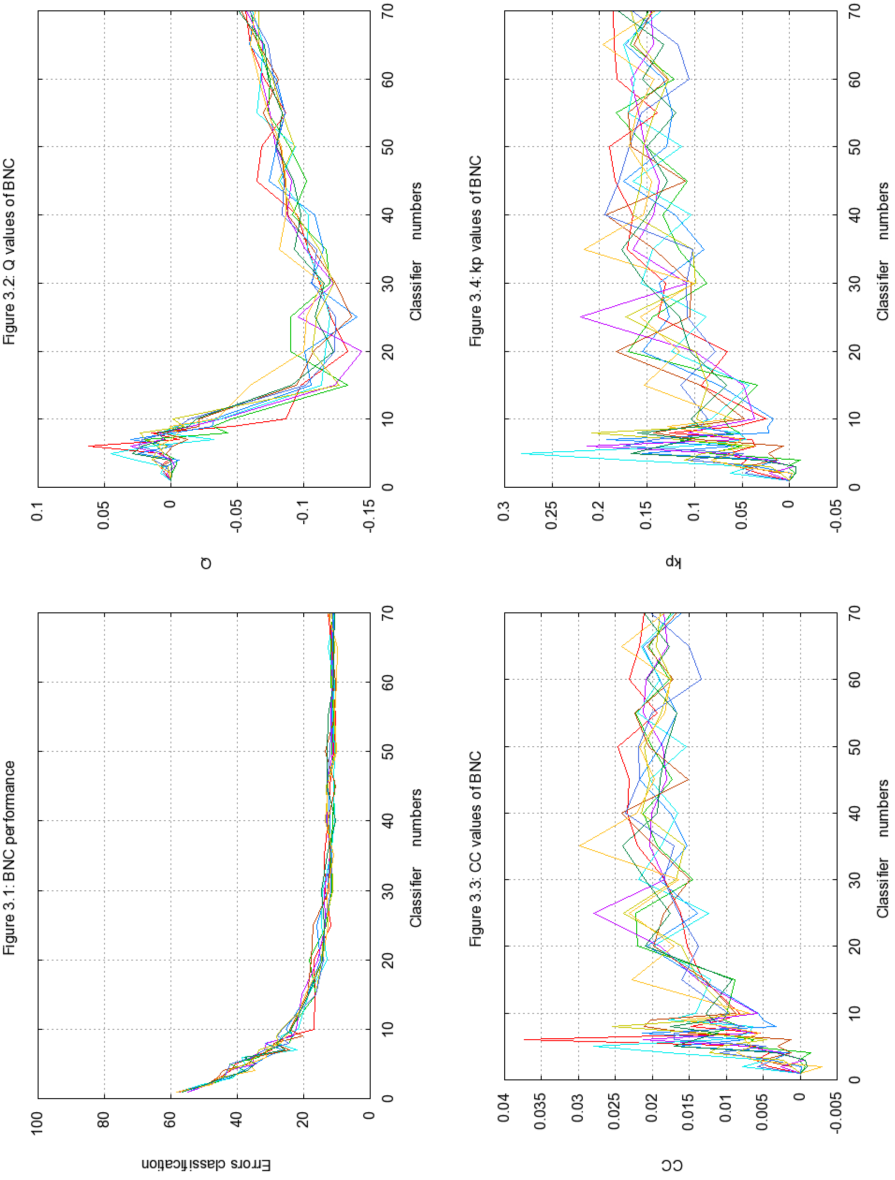


Fig. 3. Diversity and error rates of BNC on Ionosphere

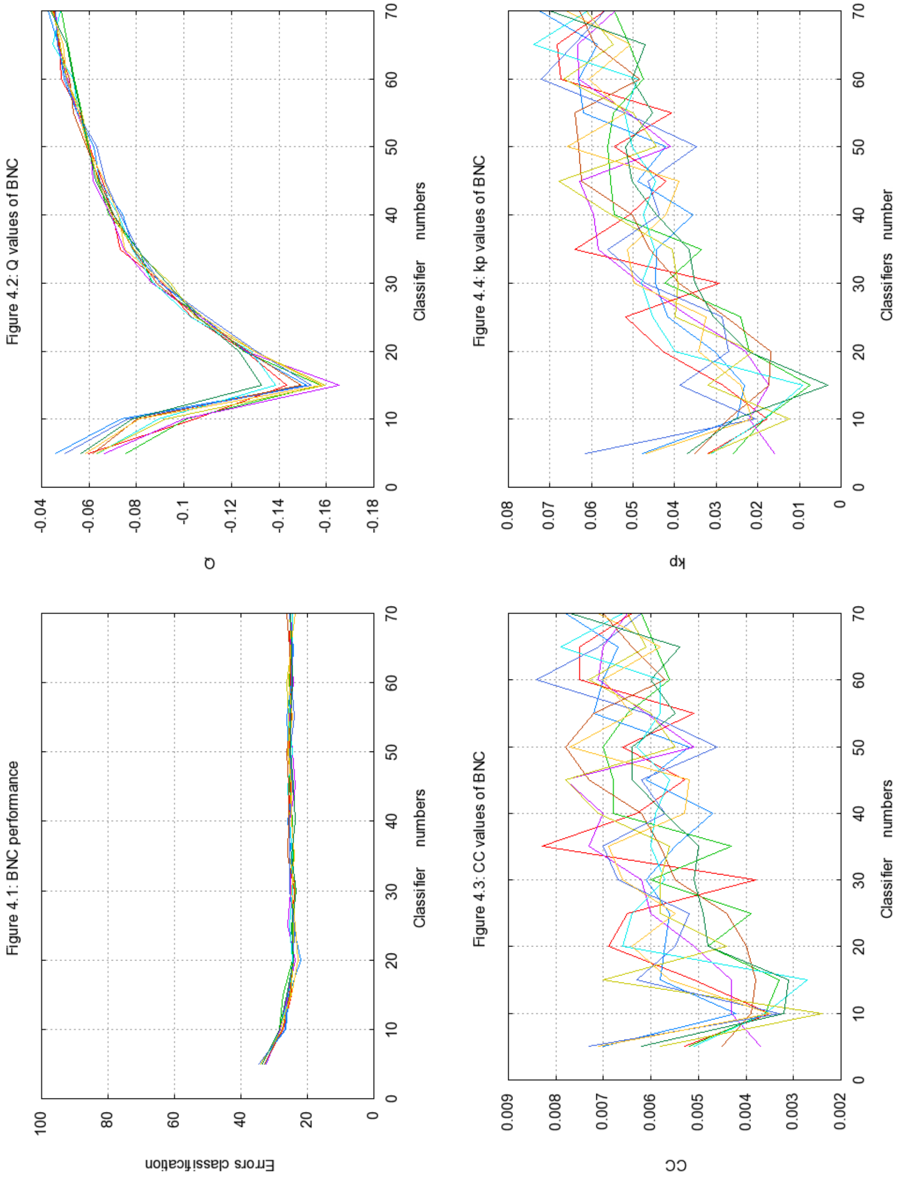


Fig. 4. Diversity and error rates of BNC on Tic Tac Toe

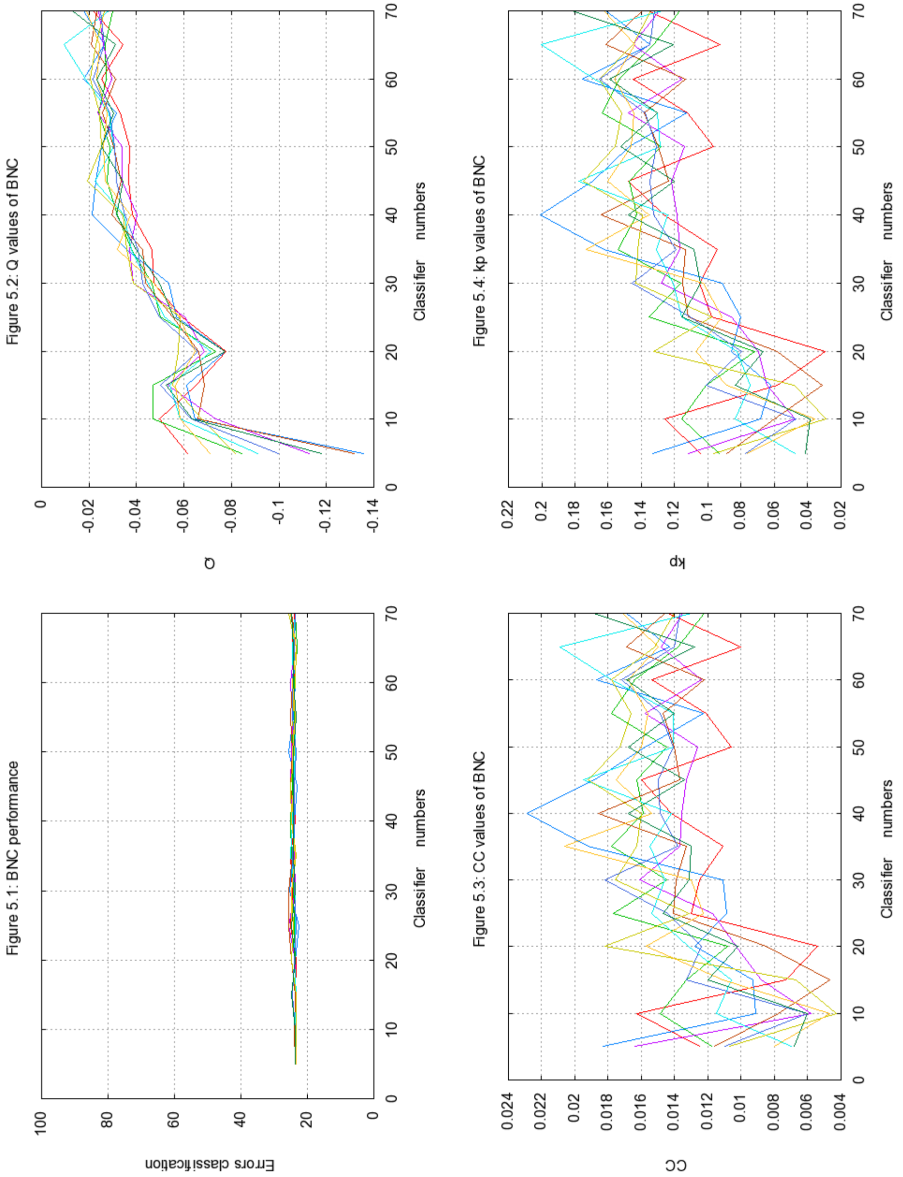


Fig. 5. Diversity and error rates of BNC on Transfusion

It consists on dividing the data sample into 10 subsets. In turn, each subset will be used for testing and the rest are assembled together for learning. Finally, the average of these 10 runs is reported.

Figures 1, 2, 3, 4 and 5 present the performance of *BNC* and the values of the 3 diversity measures on the *Credit German*, *Diabetes*, *Ionosphere*, *Tic Tac Toc* and *Transfusion* data sets respectively.

In figure 1.1, we remark that the performance of *BNC* starts to stabilize when using ensembles of 20 classifiers, for high diversity data (*DD*=98.59%). The classifiers generated are negatively depend ($\mathcal{Q} \leq$ -0.02). The minimum values of $\mathcal{Q}$ *Statistic* are obtained with classifier numbers varying between 10 and 20. From figure 1.3 and figure 1.4, the values of the 2 measures *CC* and *Kp* are very divers and the variation curves are ascending, while the curve of the values of Q is upward then downward.

In figure 2.1, the best performance of *BNC* is obtained with divers classifier ensembles (with $\mathcal{Q}$=-0.3 as minimum average values). In figure 2.2, the minimum values of $\mathcal{Q}$ *Statistic* are obtained with 20 classifiers. For *Diabetes* data (*DD*=22.83%), there is a relation between $\mathcal{Q}$ *Statistic* and the *BNC* performance.

With high divers data (figure 3.2), the first generated classifiers are independent but the rest are negatively depend ($\mathcal{Q} \leq$ -0.15). The minimum values of $\mathcal{Q}$ *Statistic* are obtained with classifier numbers varying between 15 and 30. With less than 20 classifiers, the error rate decreases about 40% (figure 3.1).

From figure 4.1, the difference between the error rates of the first classifier and the generated thereafter, is not important. This show that *Boosting* can converge to the best performance with few classifiers. For this case, Q Statistic is informative. In Figure 4.3 and 4.4, the values of Kp and CC vary an a very arbitrary way.

For the *Transfusion* data set (*DD*=1.07%), the classifier generation does not help to increase *BNC* performance. We conclude that it is not recommanded to use *AdaBoost* for this type of data.

Concerning diversity measures, we can note that for 2-class problems, the values of ρ and k_p are not correlated with *AdaBoost* performance using *nominal classifiers*. The $\mathcal{Q}$ *Statistic* seems like a good measure of model diversity that has a relationship with the performance of *AdaBoost* and then for can be used to stop classifier learning .

5 Conclusions

In this paper, we have study the diversity of *nominal classifiers* in *AdaBoost.M2*. We have compared 3 diversity measures for 2-class problems. We have found that the $\mathcal{Q}$ *Statistic* is significantly correlated with the *AdaBoost* performance, especially for very divers data sets. Then, it's possible to use this measure as a stopping criteria for ensemble learning . But for very correlated data sets, no measure is useful. This results should be confirmed with more correlated data. The diversity of data sets should then be taken into account in *AdaBoost* learning process. It's also interesting to study $\mathcal{Q}$ *Statistic* diversity to see if it can be used in *AdaBoost* for many class problems.

References

1. Aksela, M., Laaksonen, J.: Using diversity of errors for selecting members of a committee classifier. Pattern Recognition 39(4), 608–623 (2006)
2. Asuncion, A., Newman, D.: Machine Learning Repository (2007)
3. Gavin, B., Jeremy, W., Rachel, H., Xin, Y.: Diversity creation methods: A survey and categorisation. Journal of Information Fusion 6, 5–20 (2005)
4. Brown, G., Kuncheva, L.I.: "Good" and "Bad" Diversity in Majority Vote Ensembles. In: El Gayar, N., Kittler, J., Roli, F. (eds.) MCS 2010. LNCS, vol. 5997, pp. 124–133. Springer, Heidelberg (2010)
5. Freund, Y.: Boosting a weak learning algorithm by majority. Information and Computation 121, 256–285 (1995)
6. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: 13th International Conference on Machine Learning, Bari, Italy (1996)
7. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)
8. Gacquer, D., Delcroix, V., Delmotte, F., Piechowiak, S.: On the Effectiveness of Diversity When Training Multiple Classifier Systems. In: Sossai, C., Chemello, G. (eds.) ECSQARU 2009. LNCS, vol. 5590, pp. 493–504. Springer, Heidelberg (2009)
9. Ko, A.H.R., Sabourin, R., Soares de Oliveira, L.E., de Souza Britto, A.: The implication of data diversity for a classifier-free ensemble selection in random subspaces. In: International Conference on Pattern Recognition, pp. 1–5 (2008)
10. Kohavi, R.: A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In: Actes d'International Joint Conference on Artificial Intelligence, pp. 1137–1143 (1995)
11. Kuncheva, L.I., Skurichina, M., Duin, R.P.W.: An experimental study on diversity for bagging and boosting with linear classifiers. Information Fusion 3(4), 245–258 (2002)
12. Kuncheva, L.I., Rodriguez, J.J.: Classifier Ensembles for fMRI Data Analysis: An Experiment. Magnetic Resonance Imaging 28(4), 583–593 (2010)
13. Meddouri, N., Maddouri, M.: Adaptive Learning of Nominal Concepts for Supervised Classification. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) KES 2010. LNCS, vol. 6276, pp. 121–130. Springer, Heidelberg (2010)
14. Shipp, C.A., Kuncheva, L.I.: An investigation into how adaboost affects classifier diversity. In: Proc. of Information Processing and Management of Uncertainty in Knowledge-Based Systems, pp. 203–208 (2002)

Extreme Value Prediction for Zero-Inflated Data

Fan Xin[1] and Zubin Abraham[2]

[1] Department of Statistics, Michigan State University
[2] Department of Computer Science & Engineering, Michigan State University
{fanxin,abraha84}@msu.edu

Abstract. Depending on the domain, there may be significant ramifications associated with the occurrence of an extreme event (for e.g., the occurrence of a flood from a climatological perspective). However, due to the relative low occurrence rate of extreme events, the accurate prediction of extreme values is a challenging endeavor. When it comes to zero-inflated time series, standard regression methods such as multiple linear regression and generalized linear models, which emphasize estimating the conditional expected value, are not best suited for inferring extreme values. And so is the case when the the conditional distribution of the data does not conform to the parametric distribution assumed by the regression model. This paper presents a coupled classification and regression framework that focuses on reliable prediction of extreme value events in a zero-inflated time series. The framework was evaluated by applying it on a real-world problem of statistical downscaling of precipitation for the purpose of climate impact assessment studies. The results suggest that the proposed framework is capable of detecting the timing and magnitude of extreme precipitation events effectively compared with several baseline methods.

1 Introduction

The notion behind being able to foretell the occurrence of an extreme event in a time series is very appealing, especially in domains with significant ramifications associated with the occurrence of an extreme events. Predicting pandemics in an epidemiological domain or forecasting natural disasters in a geological and climatic environment are examples of applications that give importance to detection of extreme events. Unfortunately, the accurate prediction of the timing and magnitude of such events is a challenge given their low occurrence rate. More so, the prediction accuracy depends on the regression method used as well as characteristics of the data. On the one hand, standard regression methods such as generalized linear model (GLM) emphasize estimating the conditional expected value, and thus, are not best suited for inferring extremal values. On the other hand, methods such as quantile regression are focused towards estimating the confidence limits of the prediction, and thus, may overestimate the frequency and magnitude of the extreme events. Though methods for inferring extreme value distributions do exist, combining them with other predictor variables for prediction purposes remains a challenging research problem.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 318–329, 2012.

Standard regression methods typically assume that the data conform to certain parametric distributions (e.g., from an exponential family). Such methods are ineffective if the assumed distribution does not adequately model characteristics of the real data. For example, a common problem encountered especially in modeling climate and ecological data is the excess probability mass at zero. Such zero-inflated data, as they are commonly known, often lead to poor model fitting using standard regression methods as they tend to underestimate the frequency of zeros and the magnitude of extreme values in the data. One way for handling such type of data is to identify and remove the excess zeros and then fit a regression model to the non-zero values. Such an approach, can be used, for example, to predict future values of a precipitation time series [13], in which the occurrence of wet or dry days is initially predicted using a classification model prior to applying the regression model to estimate the amount of rainfall for the predicted wet days. A potential drawback of this approach is that the classification and regressions models are often built independent of each other, preventing the models from gleaning information from each other to potentially improve their predictive accuracy. Furthermore, the regression methods used in modeling the zero-inflated data do not emphasize accurate prediction of extreme values.

The paper presents an integrated framework that simultaneously classifies data points as zero-valued or not, and apply quantile regression to accurately predict extreme values or the tail end of the non-zero values of the distribution by focussing on particular quantiles.

We demonstrate the efficiency of the proposed approach on modeling climate data (precipitation) obtained from the Canadian Climate Change Scenarios Network website [1]. The performance of the approach is compared with four baseline methods. The first baseline is the general linear model (GLM) with a Poisson distribution. The second baseline used is the general linear model using an exponential distribution coupled with a binomial distribution classifier(GLM-C). A zero-inflated Poisson was used as the third baseline method (ZIP). The fourth basesline was quantile regression. Empirical results showed that our proposed framework outperforms the baselines for majority of the weather stations investigated in this study.

In summary, the main contributions of this paper are as follows:

- We compare and analyze the performance of models created using variants of GLM, quantile regression and ZIP approaches to accurately predict values for extreme data points that belong to a zero-inflated distribution.
- We present a approach optimized for modeling zero-inflated data that outperforms the baseline methods in predicting the value of extreme data points.
- We successfully demonstrated the proposed approach to the real-world problem of downscaling precipitation climate data with application to climate impact assessment studies.

2 Related Work

The motivation behind the presented model is accurately predicting extreme values in the presence of zero-inflated data. Previous studies have shown that additional precautions must be taken to ensure that the excess zeros do not lead to poor fits [2] of the regression models. A typical approach to model a zero-inflated data set is to use a mixture distribution of the form $P(y|\mathbf{x}) = \alpha\pi_0(\mathbf{x}) + (1-\alpha)\pi(\mathbf{x})$, where π_0 and π are functions of the predictor variables $\mathbf{x}$ and α is a mixing coefficient that governs the probability an observation is a zero or non-zero value. This approach assumes that the underlying data are generated from known parametric distributions, for example, π may be Poisson or negative binomial distribution (for discrete data) and lognormal or Gamma (for continuous data).

Generally, simple modeling of zero values may not be sufficient, especially in the case of zero-inflated climate data such as that of precipitation where extreme value observations, (that could indicate floods, droughts, etc) need to be accurately modeled. Due to the significance of extreme values in climatology and the increasing trend in extreme precipitation events over the past few decades, a lot of work needs to be done in analysing the trends in precipitation, temperature, etc., for regions in United states, Canada, among others [3]. Katz et al. introduces the common approaches used in climate change research, especially with regard to extreme values[4].

The common approaches to modeling extreme events are based on general extreme value theory [5], Pareto distribution [10], generalized linear modeling [6], hierarchical Bayesian approaches [9], etc. Gumbel [8] and Weibull [12] are the more common variants of general extreme value distribution used. There are also Bayesian models [11] that try augmenting the model with spatial information. Watterson et al. propose a model that also deals with the skewness of non-zero data/intermittency of precipitation using gamma distribution to interpret changes in precipitation extremes [7]. In contrast, the framework presented in this paper handles the intermittency of the data by coupling a logistic regression classifier to the quantile regression part of the model.

3 Preliminaries

Consider a multivariate time series $\mathbf{L} = (\mathbf{x}_t, y_t)$, where $t \in \{1, 2, \cdots, n\}$ is a discrete-valued index for time, $\mathbf{x}_t$ is a d-dimensional vector of predictor variables at time t, and y_t is the corresponding value for the response (target) variable. Given an unlabeled sequence of multivariate observations $\mathbf{x}_\tau$, where $\tau \in \{n+1, \cdots, n+m\}$, our goal is to learn a target function $f(\mathbf{x}, \boldsymbol{\beta})$ that best estimates the values of the response variable by minimizing the expected loss $\mathcal{E}_{\mathbf{x},y}[\mathcal{L}(\mathbf{y}, f(\mathbf{x}, \boldsymbol{\beta}))]$. The weight vector $\boldsymbol{\beta}$ denotes the regression coefficients to be estimated from the training data $\mathbf{L}$.

Multiple linear regression (MLR) is one of most widely used regression methods due to its simplicity. It assumes $f(\mathbf{x}, \boldsymbol{\beta}) = \boldsymbol{\beta}^T\mathbf{x}$ (where $\mathbf{x}$ is a $(d+1)$-dimensional vector whose first element $x_0 = 1$ and $\boldsymbol{\beta} \in \Re^{d+1}$ is the weight vector) and the response variable y is related to $f(\mathbf{x}, \boldsymbol{\beta})$ via the following equation:

$$y = \boldsymbol{\beta}^T\mathbf{x} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(0, \sigma^2).$$

As a result, $P(y|\mathbf{x}) \sim N(\boldsymbol{\beta}^T\mathbf{x}, \sigma^2)$ and $\mathcal{E}_{y|\mathbf{x}}[y] = \int yP(y|\mathbf{x})dy = \boldsymbol{\beta}^T\mathbf{x}$. Since the predicted value of the response variable for a test data point $\mathbf{x}_\tau$ is $\boldsymbol{\beta}^T\mathbf{x}_\tau$, this implies that the predictions made by MLR focus primarily on the average value of y given $\mathbf{x}_\tau$. This explains the limitation of MLR in terms of inferring extreme values in a given time series. The parameter vector $\boldsymbol{\beta}$ in MLR can be estimated using the maximum likelihood (ML) approach to obtain

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y},$$

where $\mathbf{X}$ is the $n \times (d+1)$ design matrix and $\mathbf{y}$ is an $n \times 1$ column vector for the observed values of the response variable.

The drawback of simple linear regression is that it is built on a strong assumption -namely, normality. Unfortunately, real world data may not always have a normal distribution and may be skewed to one side or may not cover the whole range of real numbers or may have a heavier tail than the normal distribution, etc. Hence, alternative approaches that are not constrained by such assumptions such as GLM may be used.

3.1 Generalized Linear Model(GLM) and 2-Step GLM (GLM-C)

The generalized linear model is one of most widely used regression methods due to its simplicity. Generally, a GLM consists of three elements:

1. The response variable $\mathbf{Y}$, which has a probability distribution from the exponential family.
2. A linear predictor $\eta = \mathbf{X}\boldsymbol{\beta}$
3. A link function $g(\cdot)$ such that $E(\mathbf{Y}|\mathbf{X}) = \boldsymbol{\mu} = g^{-1}(\eta)$

where, $\mathbf{Y} \in \mathcal{R}^{n\times 1}$ is the response variables vector, $\mathbf{X} \in \mathcal{R}^{n\times d}$ is the design matrix with all 1 in the last column. $\boldsymbol{\beta} \in \mathcal{R}^{p\times 1}$ is the parameter vector. Since the link function shows the relationship between the linear predictor and the mean of the distribution, it is very important to understand the detail about the data before arbitrarily using the canonical link function. In our case, since the precipitation data are always non-negative and values represented using a millimeter scale, the non-zero data may be treated as count data allowing us to use Poisson distribution or an exponential distribution to describe the data. Hence, in our experiments we always choose $\log(\cdot)$ as the link function and choose to use Poisson distribution. We scale the Y used in the regression model to be $10 \times Y$:

$$(10 \times Y_i)|X_i \sim Poi(\lambda_i)$$

$$E((10 \times Y_i)|X_i) = \lambda_i = g^{-1}(\eta_i) = g^{-1}(X_i\beta);$$

The histogram in Figure 1 is a representation of the data belonging to station-1. It is clear that the number of zero is too large. The second histogram which is without zero looks similar to a kind of Poisson or exponential distribution.

Considering the large number of zeros, one is motivated to perform classification first to eliminate the zero values before any regression. There are many

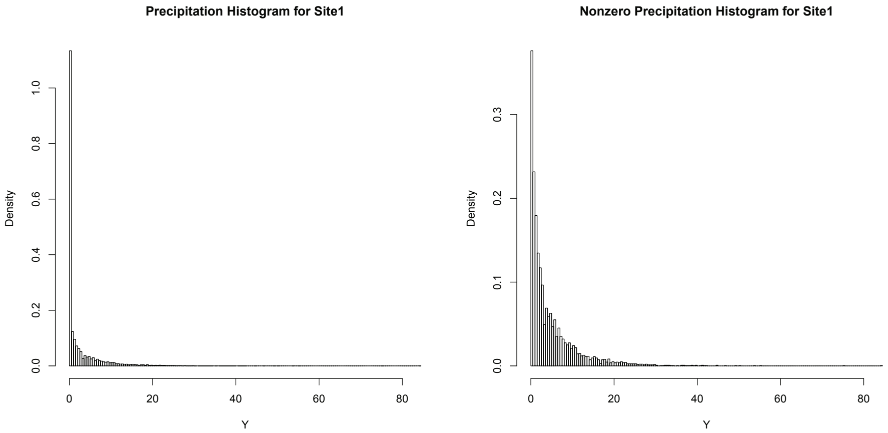


Fig. 1. Comparison of the histogram of the original distribution of data at Station-1 with its truncated counterpart

classification methods available. But for the purpose of our experiments, we use logistic regression (which is also a variation of GLM) to do the classification. The response variable Y^* of logistic regression is a binary variable defined as:

$$Y^* = \begin{cases} 1 & Y > 0, \\ 0 & Y = 0 \end{cases}$$

The detail of the model is as follows: The link function is a logit link $g(p) = \log(\frac{p}{1-p})$, such that,

$$Y_i^*|X_i \sim Bin(p_i)$$

$$E(Y_i^*|X_i) = p_i = g^{-1}(\eta_i) = g^{-1}(X_i\beta);$$

When we derive the fitted values, they will be transferred to be binary:

$$f^* = \begin{cases} 1 & 1 \geq \hat{Y}^* > 0.5, \\ 0 & 0.5 \geq \hat{Y}^* \geq 0 \end{cases}$$

The second part is a GLM with exponential distribution, the response variable Y' is just those non-zero data, and the link function is $g(\cdot) = \log(\cdot)$:

$$Y_i'|X_i \sim Exp(\lambda_i)$$

$$E(Y_i'|X_i) = \lambda_i = g^{-1}(\eta_i) = g^{-1}(X_i\beta);$$

Then, we got fitted-value $\mathbf{f}'$ for all X_i
Finally, we report the product of those two fitted-values $\hat{\mathbf{Y}} = \mathbf{f}^* \times \mathbf{f}'$

To fit the GLM model, we use iteratively reweighted least squares(IRLS) method for maximum likelihood estimation of the model parameters.

3.2 Zero Inflated Poisson Regression(ZIP)

Differing from the methods above, zero inflated poisson regression treats the zero as a mixture of two distributions: a Bernoulli distribution with probability π_i to get 0, and a Poisson distribution with parameter μ (let $Pr(\cdot;\mu)$ denote the probability density function). In fact, the ZIP regression model is defined as:

$$Pr(Y = y_i|x_i) = \begin{cases} \pi_i + (1-\pi_i)Pr(Y_i = 0; \lambda_i) & y_i = 0, \\ (1-\pi_i)Pr(Y = y_i; \lambda_i) & y_i > 0 \end{cases}$$

where $0 < \pi_i < 1$, and

$$\text{logit}(\pi_i) = \log(\frac{\pi_i}{1-\pi_i}) = x_i\boldsymbol{\beta}_1$$

$$\log(\mu_i) = x_i\boldsymbol{\beta}_2$$

where $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$ are all regression parameter. Both of them could be found by maximizing the likelihood function. For the purpose of the experiments, we used the R package 'pscl' to fit the model.

3.3 Quantile Linear Regression(QR) and 2-step QR(QR-C)

Quantile regression was used to estimate the specified quantile of a population. Hence, if the objective of the regression is to estimate the conditional quantile(e.g., median) of $\mathbf{Y}$ instead of a conditional mean like MLR and Ridge regression, one may use quantile regression. Its loss function for the linear regression model is:

$$f(\mathbf{b}) = \sum_{i=1}^{N} \rho_\tau(Y_i - \mathbf{X}_i^T\mathbf{b}), \text{ and } \hat{\boldsymbol{\beta}} = \arg\min_{\mathbf{b}} f(\mathbf{b}),$$

where

$$\rho_\tau(u) = \begin{cases} \tau u & u > 0 \\ (\tau - 1)u & u \le 0 \end{cases}$$

Let $F_Y(y) = P(Y \le y)$ be the distribution function of a real valued random variable Y. The τ^{th} quantile of Y is given by:

$$Q_Y(\tau) = F^{-1}(\tau) = \inf\{y : F_Y(y) \ge \tau\}$$

It can be proved that the $\hat{y}$ which minimizes $E\rho_\tau(y - \hat{y})$ should satisfy that $F_Y(\hat{y}) = \tau$. Thus, quantile regression will find the τ^{th} quantile of a random variable, for example:

$$\text{Median}(\mathbf{Y}|\mathbf{X}) = X\hat{\boldsymbol{\beta}}^{qr}; \hat{\boldsymbol{\beta}}^{qr} = \arg\min_{\mathbf{b}} \sum \rho_{0.5}(y_i - \mathbf{X_i^T b})$$

For the purpose of the experiments conducted, we always used $\tau = 0.95$ to represent extreme high value. Unlike the least squares methods mentioned above which could be solved by numerical linear algebra, the solution to quantile regression is relatively non-trivial. Linear programming is used to solve the loss function by converting the problem to the following form.

$$\min_{\mathbf{u},\mathbf{v},\mathbf{b}} \{\tau \mathbf{e}_{\mathbf{N}}^{\mathbf{T}} \mathbf{u} + (1-\tau)\mathbf{e}_{\mathbf{N}}^{\mathbf{T}} \mathbf{v} | \mathbf{Y} - \mathbf{X}\mathbf{b} = \mathbf{u} - \mathbf{v}; \mathbf{b} \in \mathcal{R}^p; \mathbf{u}, \mathbf{v} \in \mathcal{R}_+^N\}$$

For the same reason as mentioned in the Section 3.1, a classification method should be incorporated along with the regression model. We used logistic regression for classification, and quantile regression on those nonzero Y. Finally, we report the product of those two fitted values. Quantile regression may return a negative value, which we force to 0. We do this because precipitation is always non-negative.

4 Framework for Integrated Classification and Regression

Now that we have introduced quantile regression, which is an integral part of our objective function we will elaborate the motivation behind the various components of the proposed objective function. Since zero-inflated data is best described with the help of a classifier that help identify non-zero values and a regression component to address non-zero values, our framework consists of both components. For the classifier component we use least square support vector machine and for the regression component, we use the intuition of quantile regression to help focus the regression of extreme values. Since the final prediction of the data point using this framework is a product of the regression and classification component, the quantile regression component is built to work on the eventual predicted return value, thereby integrating both the classifier and regression components.

4.1 Integrated Classifier and Regression for Extreme Values(ICRE)

The classification and regression models developed in this study are designed to minimize the following objective function:

$$\begin{aligned} \arg\min_{\boldsymbol{\omega}_1,\boldsymbol{\omega}_2} L(\boldsymbol{\omega}_1,\boldsymbol{\omega}_2) &= \frac{1}{n}\sum_{i=1}^{n}(1-(2y_i-1)f_i)^2 \\ &+ \frac{1}{n^*}\sum_{i=1}^{n} y_i \rho_\tau (y_i' - f_i' \times (f_i+1)/2) + \lambda(||\boldsymbol{\omega}_1||^2 + ||\boldsymbol{\omega}_2||^2) \end{aligned} \tag{1}$$

where n^* is the number of nonzero y_i. Then it can be expanded as follows:

$$\begin{aligned} \arg\min_{\boldsymbol{\omega}_1,\boldsymbol{\omega}_2} L(\boldsymbol{\omega}_1,\boldsymbol{\omega}_2) &= \frac{1}{n}\sum_{i=1}^{n}(1-(2y_i-1)(\mathbf{x}_i^T\boldsymbol{\omega}_2))^2 \\ &+ \frac{1}{n^*}\sum_{i=1}^{n} y_i \rho_\tau (y_i' - (\mathbf{x}_i^T\boldsymbol{\omega}_1) \times (\mathrm{sign}((\mathbf{x}_i^T\boldsymbol{\omega}_2 + 1)/2))) \\ &+ \lambda(||\boldsymbol{\omega}_1||^2 + ||\boldsymbol{\omega}_2||^2) \end{aligned} \tag{2}$$

The rationale for the design of our objective function is as follows. The first term which corresponds to the regression part of the equation represents quantile regression performed for only the observed non-zero values in the time series. The regression model is therefore biased towards estimating the non-zero extreme values more accurately and not be adversely influenced by the over-abundance of zeros in the time series. The product $f_i' \times (f_i + 1)/2$ in the first term, corresponds to the predicted output of our joint classification and regression model. The second term in the objective function, which is the main classification component, is equivalent to the least square support vector machine. And the last two terms in the objective function are equivalent to the L_2 norm used in ridge regression models to shrink the coefficients in $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$.

We consider each data point to be a representative reading at an instance of time $t \in \{1, 2, \cdots, n\}$ in the time series. Each predictor variable is standardized by subtracting its mean value and then dividing by its corresponding standard deviation. The standardization of the variables is needed to account for the varying scales.

The optimization method used while performing experiments is 'L-BFGS-B', described by Byrd et. al. (1995). It is a limited memory version of BFGS methods. This method does not store a Hessian matrix, just a limited number of update steps for it, and then it uses derivative information. Since our model includes a quantile regression component, which is not differentiable, this method of optimization is well suited to our objective function.

To solve the objective function, we used the inverse logistic function of $\mathbf{x}_i^T \boldsymbol{\omega}_2$ instead of $\text{sign}((\mathbf{x}_i^T \boldsymbol{\omega}_2 + 1)/2))$. The decision was motivated by the fact that the optimizer tries to do a line search along the steepest descent direction and finds the positive derivative along this line, which would result in a nearly flat surface for the binary component. Hence conversion of the binary report to an inverse logistic function of $\mathbf{x}_i^T \boldsymbol{\omega}_2$ was used to address this issue. During the prediction stage, we use the binary-fitted values from the SVM component.

5 Experimental Evaluation

In this section, the climate data that are used to downscale precipitation is described. This is followed by the experiment setup. Once the dataset is introduced, we analyzed the behavior of baseline models and contrasted them with ICRE in terms of relative performance of the various models when applied to this real world dataset to forecast future values of precipitation.

5.1 Data

All the algorithms were run on climate data obtained for 29 weather stations in Canada, from the Canadian Climate Change Scenarios Network website [1]. The response variable to be regressed (downscaled), corresponds to daily precipitation values measured at each weather station. The predictor variables correspond to 26 coarse-scale climate variables derived from the NCEP Reanalysis data set and the H3A2a data set(computer generated simulations), which include measurements of airflow strength, sea-level pressure, wind direction, vorticity, and humidity. The predictor variables used for training were obtained from the NCEP Reanalysis data set while the predictor variables used for the testing were obtained from the H3A2a data set. The data span a 40-year period, 1961 to 2001. The time series was truncated for each weather station to exclude days for which temperature or any of the predictor values are missing.

5.2 Experimental setup

The first step was to standardize the predictor variables by subtracting its mean value and then dividing by its corresponding standard deviation to account for their varying scales. The training size used was 10yrs worth of data and the test size, 25yrs. During the validation process, the selection of the parameter λ was done using the score returned by RMSE-95. Also, to ensure the experiments replicated the real world scenario where the prediction for a future timeseries needs to be performed using simulated values of the predictor variables for the future time series, we used simulated values for the corresponding predictor variables obtained from H3A2a climate scenario as $\mathbf{X}_U$, while $\mathbf{X}_L$ are values obtained from NCEP. All the experiments were run for 37 stations.

5.3 Baseline Algorithm

We compare the performance of ICRE with baseline models created using general linear model(GLM), general linear model with classification (GLM-C), quantile regression(QR), quantile regression with classification and zero-inflated Poisson(ZIP). Further details about the baselines are provided below.

General Linear Model (GLM). The baseline GLM refers to the generalized linear model that uses a Poisson distribution as a link function, resulting in the regression function $\log(\lambda) = X\beta$, where $E(Y|X) = \lambda$

General Linear Model with Classification (GLM-C). Unlike the previous baseline (GLM), GLM-C refers to a two step generalized linear model that uses a Binomial distribution, for the classifier with the model described as $logit(p) = X\beta$, and $E(Y' = 1|X) = p$ which $Y' = 1$ when $Y > 0$ and $Y' = 0$ when $Y = 0$ and a second step that uses a generalized linear model with an exponential distribution that is built only on non-zero response data points. The regression function is $\log(\lambda) = X\beta$, which $E(Y|X) = \lambda$. The eventual predicted value for each data point is the product of the two respective fitted values.

Quantile Regression (QR). The baseline QR refers to the regular quantile regression described earlier in the preliminary section 3

Quantile Regression with Classification(QR-C). The baseline QR-C refers to a two step model that has a GLM that uses a binomial distribution that acts as a classifier and a regular quantile regression model that is built on non-zero valued data points as described earlier in the preliminary section. These two models that comprise QR-C are built independent of each other and the eventual predicted value for each data point is the product of the two respective fitted values.

Zero Inflated Poisson(ZIP). Zero Inflation Poisson model used as a baseline and is similar to the ZIP model described in Section 3.

5.4 Evaluation Criteria

The motivation behind the selection of the various evaluation metrics was to evaluate the different algorithms in terms of predicting the magnitude and the timing of the extreme events.The following criteria to evaluate the performance of the models are used:

- Root Mean Square Error (RMSE), which measures the difference between the actual and predicted values of the response variable, i.e.:
  RMSE $= \sqrt{\frac{\sum_{i=1}^{n}(y'_i - f'_i f_i)^2}{n}}$.
- RMSE-95, which we use to measure the difference between the actual and predicted value of the response variable for only the extreme data points(j). Extreme data points refer to the points whose actual value were 95 percentile and above. The equation is with respect to 95 percentile, as throughout this paper, we associate data points that are 95 percentile and above as extreme values, i.e.:
  RMSE-95 $= \sqrt{\frac{\sum_{j=1}^{n/20}(y'_j - f_i f'_j)^2}{n/20}}$.
- Confusion matrices will be computed to visualize the precision and recall of extreme and non-extreme events. F-measure, which is the harmonic mean between recall and precision values will be used as a score that evaluates the precision and recall results.
  F-measure $= \frac{2 \times Recall \times Precision}{Recall + Precision}$

To summarize, RMSE-95 is used for measuring magnitude and F-measure measures the correctness of the timing of the extreme events.

5.5 Experimental Results

The results section consists of two main sets of experiments. The first set of experiments evaluates the impact of zero-inflated data on modeling extreme values. The second section compares the performance of ICRE with the baseline methods which are followed .

Impact of Zero-Inflated Data on Extreme Value Prediction. Unlike regular data which may be modeled using regression, modeling zero-inflated data usually involves a classifier and a regression component. The classifier is used to identify zero and non-zero values, which is followed by regression for the non-zero values. But since the focus of the paper is on extreme data points within zero-inflated data, the impact of the classifier is unclear. In this section, we compare the impact of including the classifier in modeling extreme values of zero-inflated data. We compared QR with QR-C and GCM with GCM-C and show the results in Table 1. Note that the percentage of wins for F-measure, recall, precision may not total to 100 in the case of a tie.

Table 1. Percentage of stations won

	QR-C	QR	GLM-C	GLM
RMSE-95	0	100	67.57	32.43
F-Measure	81.08	18.92	18.92	35.13

As shown in the Table 1, it isn't clear that using an independent classifier along with regression for modeling extreme values among zero inflated data is preferred. But the results do indicate that the inclusion or exclusion of a classifier with the regression model built independent of each other may compromise either RMSE-95 (by overestimating the magnitude) or F-measure (mistiming predicting an extreme value), without necessarily compromising both together.

Comparison of ICRE to Baseline Methods. Table 2 shows the relative performance of ICRE to all the baseline methods in terms of percentage of stations outperformed against the baseline method in terms of RMSE-95 values calculated on extreme rain days. In terms of RMSE of extreme rain days, as shown in Table 2, ICRE outperformed the baselines (except QR) in almost every one of the 37 stations. But QR was the best across all methods for RMSE-95 of extreme days. In terms of F-measure that was computed based on recall and precision of identifying extreme events, ICRE again outperformed the baselines(except QR-C) in majority of the 37 stations. But ICRE was only able to outperform QR-C in 16 or the 37 stations in terms of F-measure. Although QR performed the best in terms of estimating magnitude for those extreme events, it over-estimate the timing of the events as seen by the relatively lower F-measure score. QR-C did the reverse, it did reasonably well in terms of modeling the timing, but performed very poorly in terms of the magnitude of the events by overestimating.

Table 2. Percentage of stations ICRE outperformed the baseline

	QR-C	QR	GLM-C	GLM	ZIP
RMSE-95	91.89	0	97.3	97.3	97.3
F-Measure	43.24	62.16	89.19	89.19	91.9

6 Conclusions

This paper compare and analyze the performance of models created using variants of GLM, quantile regression and ZIP approaches to accurately predict values for extreme data points that belong to a zero-inflated distribution. An alternate framework(ICRE) was present that outperforms the baseline methods and the effectiveness of the model was demonstrated on climate data to predict the amount of precipitation at a given station. For future work, we plan to extend the framework to a semi-supervised setting.

References

1. Canadian Climate Change Scenarios Network, Environment Canada, `http://www.ccsn.ca/`
2. Ancelet, S., Etienne, M.-P., Benot, H., Parent, E.: Modelling spatial zero-inflated continuous data with an exponentially compound poisson process. Environmental and Ecological Statistics (April 2009), doi:10.1007/s10651-009-0111-6
3. Kunkel, E.K., Andsager, K., Easterling, D.: Long-Term Trends in Extreme Precipitation Events over the Conterminous United States and Canada. J. Climate, 2515–2527 (1999)
4. Katz, R.: Statistics of extremes in climate change. Climatic Change, 71–76 (2010)
5. Gaetan, C., Grigoletto, M.: A hierarchical model for the analysis of spatial rainfall extremes. Journal of Agricultural, Biological, and Environmental Statistics (2007)
6. Clarke, R.T.: Estimating trends in data from the Weibull and a generalized extreme value distribution. Water Resources Research (2002)
7. Watterson, I.G., Dix, M.R.: Simulated changes due to global warming in daily precipitation means and extremes and their interpretation using the gamma distribution. Journal of Geophysical Research (2003)
8. Booij, M.J.: Extreme daily precipitation in Western Europe with climate change at appropriate spatial scales. International Journal of Climatology (2002)
9. Ghosh, S., Mallick, B.: A hierarchical Bayesian spatio-temporal model for extreme precipitation events. Environmetrics (2010)
10. Dorland, C., Tol, R.S.J., Palutikof, J.P.: Vulnerability of the Netherlands and Northwest Europe to storm damage under climate change. Climatic Change, 513–535 (1999)
11. Cooley, D., Nychka, D., Naveau, P.: Bayesian spatial modeling of extreme proecipitation return levels. Journal of the American Statistical Association, 824–840 (2007)
12. Clarke, R.T.: Estimating trends in data from the Weibull and a generalized extreme value distribution. Water Resources Research (2002)
13. Wilby, R.L.: Statistical downscaling of daily precipitation using daily airflow and seasonal teleconnection. Climate Research 10, 163–178 (1998)

Learning to Diversify Expert Finding with Subtopics*

Hang Su[1], Jie Tang[2], and Wanling Hong[1]

[1] School of Software, Beihang University, China, 100191
{suhang,wanling}@sse.buaa.edu.cn

[2] Department of Computer Science and Technology, Tsinghua University, China, 100084
jietang@tsinghua.edu.cn

Abstract. Expert finding is concerned about finding persons who are knowledgeable on a given topic. It has many applications in enterprise search, social networks, and collaborative management. In this paper, we study the problem of diversification for expert finding. Specifically, employing an academic social network as the basis for our experiments, we aim to answer the following question: Given a query and an academic social network, how to diversify the ranking list, so that it captures the whole spectrum of relevant authors' expertise? We precisely define the problem and propose a new objective function by incorporating topic-based diversity into the relevance ranking measurement. A learning-based model is presented to solve the objective function. Our empirical study in a real system validates the effectiveness of the proposed method, which can achieve significant improvements (+15.3%-+94.6% by MAP) over alternative methods.

1 Introduction

Given a coauthor network, how to find the top-*k* experts for a given query q? How to diversify the ranking list so that it captures the whole spectrum of relevant authors' expertise? Expert finding has long been viewed as a challenging problem in many different domains. Despite that considerable research has been conducted to address this problem, e.g., [3,17], the problem remains largely unsolved. Most existing works cast this problem as a web document search problem, and employ traditional relevance-based retrieval models to deal with the problem.

Expert finding is different from the web document search. When a user is looking for expertise collaborators in a domain such as "data mining", she/he does not typically mean to find *general* experts in this field. Her/his intention might be to find experts on different *aspects* (subtopics) of data mining (e.g., "association rules", "classification", "clustering", or "graph mining"). Recently, diversity already becomes a key factor to address the uncertainty and ambiguity problem in information retrieval [12,21]. However, the diversification problem is still not well addressed for expert finding. In this paper, we try to give an explicit diversity-based objective function for expert finding, and to leverage a learning-based algorithm to improve the ranking performance.

* The work is supported by the Natural Science Foundation of China (No. 61073073) and Chinese National Key Foundation Research (No. 60933013), a special fund for Fast Sharing of Science Paper in Net Era by CSTD.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 330–341, 2012.

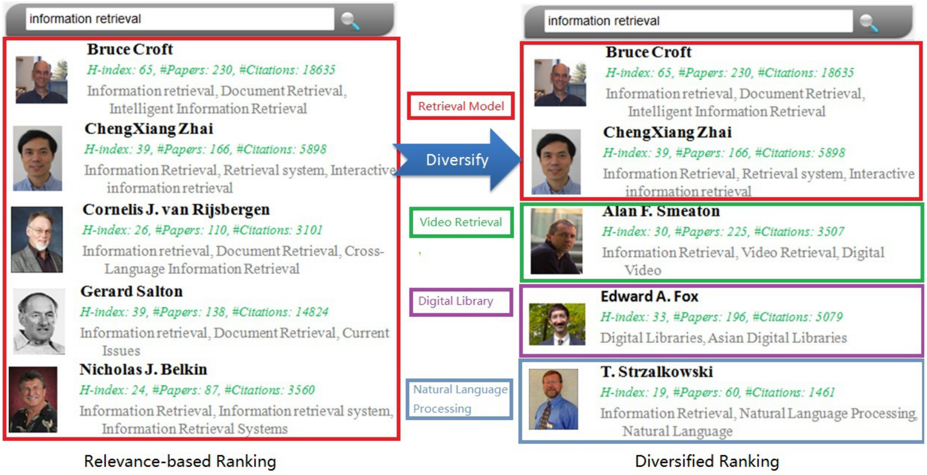

information retrieval
Bruce Croft
H-index: 65, #Papers: 230, #Citations: 18635
Information retrieval, Document Retrieval, Intelligent Information Retrieval
ChengXiang Zhai
H-index: 39, #Papers: 166, #Citations: 5898
Information Retrieval, Retrieval system, Interactive information retrieval
Cornelis J. van Rijsbergen
H-index: 26, #Papers: 110, #Citations: 3101
Information retrieval, Document Retrieval, Cross-Language Information Retrieval
Gerard Salton
H-index: 39, #Papers: 138, #Citations: 14824
Information retrieval, Document Retrieval, Current Issues
Nicholas J. Belkin
H-index: 24, #Papers: 87, #Citations: 3560
Information Retrieval, Information retrieval system, Information Retrieval Systems
Relevance-based Ranking
Retrieval Model
Diversify
Video Retrieval
Digital Library
Natural Language Processing
information retrieval
Bruce Croft
H-index: 65, #Papers: 230, #Citations: 18635
Information retrieval, Document Retrieval, Intelligent Information Retrieval
ChengXiang Zhai
H-index: 39, #Papers: 166, #Citations: 5898
Information Retrieval, Retrieval system, Interactive information retrieval
Alan F. Smeaton
H-index: 30, #Papers: 225, #Citations: 3507
Information Retrieval, Video Retrieval, Digital Video
Edward A. Fox
H-index: 33, #Papers: 196, #Citations: 5079
Digital Libraries, Asian Digital Libraries
T. Strzalkowski
H-index: 19, #Papers: 60, #Citations: 1461
Information Retrieval, Natural Language Processing, Natural Language
Diversified Ranking

Fig. 1. Illustrative example of diversified expert finding. The query is “information retrieval”, and the list on the left side is obtained by language model. All the top five are mainly working on retrieval models. The right list is obtained by the proposed diversified ranking method with four subtopics (indicated by different colors).

Motivating Examples. To illustrate this problem, Figure 1 gives an example of diversified expert finding. The list on the left is obtained by using language model, a state-of-the-art relevance-based ranking model [2]. We see that all the top five experts are mainly working on information retrieval models. The right list is obtained using the proposed diversified ranking method with four subtopics. The top two experts are working on information retrieval models, but the third one is working on multimedia retrieval, the fourth is about digital library, and the fifth is about information retrieval using natural language processing. The diversified ranking list is more useful in some sense: the user can quickly gain the major subtopics of the query, and could refine the query according to the subtopic that she/he is interested in. Additionally, the user can have the hint about what the other users are recently interested in, as the ranking list is obtained by learning from the user feedback (e.g., users’ click data).

We aim to conduct a systematic investigation into the problem of diversifying expert finding with subtopics. The problem is non-trivial and poses a set of unique challenges. First, how to detect subtopics for a given query? Second, how to incorporate the diversity into the relevance-based ranking score? Third, how to efficiently perform the expert ranking algorithm so that it can be scaled up to handle large networks?

Contributions. We show that incorporating diversity into the expert finding model can significantly improve the ranking performance (+15.3%-+94.6% in terms of MAP) compared with several alternative methods using language model, topic model and random walk. In this work, we try to make the following contributions:

– We precisely formulate the problem of diversified expert finding and define an objective function to explicitly incorporate the diversity of subtopics into the relevance ranking function.

- We present a learning-based algorithm to solve the objective function.
- We evaluate the proposed method in a real system. Experimental results validate its effectiveness.

Organization. Section 2 formulates the problem. Section 3 explains the proposed method. Section 4 presents experimental results that validate the effectiveness of our methodology. Finally, Section 5 reviews related work and Section 6 concludes.

2 Problem Definition

In this section, we formulate the problem in the context of academic social network to keep things concrete, although adaption of this framework to expert finding in other social-network settings is straightforward.

Generally speaking, the input of our problem consists of (1) the results of any topic modeling such as predefined ontologies or topic cluster based on pLSI [9] or LDA [5] and (2) a social network $G = (V, E)$ and the topic model on authors V, where V is a set of authors and $E \subset V \times V$ is a set of coauthor relationships between authors. More precisely, we can define a topic distribution over each author as follows.

Topic distribution: In social networks, an author usually has interest in multiple topics. Formally, each user $v \in V$ is associated with a vector $\theta_v \in \mathbb{R}^T$ of T-dimensional topic distribution ($\sum_z \theta_{vz} = 1$). Each element θ_{vz} is the probability (i.e., $p(z|v)$) of the user on topic z.

In this way, each author can be mapped onto multiple related topics. In the meantime, for a given query q, we can also find a set of associated topics (which will be depicted in detail in §3). Based on the above concepts, the goal of our diversified expert finding is to find a list of experts for a given query such that the list can maximally cover the associated topics of the query q. Formally, we have:

Problem 1. **Diversified Expert Finding.** Given (1) a network $G = (V, E)$, (2) T-dimensional topic distribution $\theta_v \in \mathbb{R}^T$ for all authors v in V, and (3) a metric function $f(.)$, the objective of diversified expert finding for each query q is to maximize the following function:

$$\sum_{z=1}^{T} f(k|z, G, \Theta, q) \times p(z|q) \tag{1}$$

where $f(k|z, G, \Theta, q)$ measures the relevance score of top-k returned authors given topic z; we can apply a parameter τ to control the complexity of the objective function by selecting topics with larger probabilities (i.e., minimum number of topics that satisfy $\sum_z p(z|q) \geq \tau$). In an extreme case ($\tau = 1$), we consider all topics.

Please note that this is a general formulation of the problem. The relevance metric $f(k|z, G, \Theta, q)$ can be instantiated in different ways and the topic distribution can also be obtained using different algorithms. Our formulation of the diversified expert finding is very different from existing works on expert finding [3,16,17]. Existing works have mainly focused on finding relevant experts for a given query, but ignore the diversification over different topics. Our problem is also different from the learning-to-rank work [11,23], where the objective is to combine different factors into a machine learning model to better rank web documents, which differs in nature from our diversified expert finding problem.

3 Model Framework

3.1 Overview

At a high level, our approach primarily consists of three steps:

- We employ an unified probabilistic model to uncover topic distributions of authors in the social network.
- We propose an objective function which incorporates the topic-based diversity into the relevance-based retrieval model.
- We present an efficient algorithm to solve the objective function.

3.2 Topic Model Initialization

In general, the topic information can be obtained in many different ways. For example, in a social network, one can use the predefined categories or user-assigned tags as the topic information. In addition, we can use statistical topic modeling [9,5,20] to automatically extract topics from the social networking data. In this paper, we use the author-conference-topic (ACT) model [20] to initialize the topic distribution of each user. For completeness, we give a brief introduction of the ACT model. For more details, please refer to [20].

ACT model simulates the process of writing a scientific paper using a series of probabilistic steps. In essence, the topic model uses a latent topic layer $Z = \{z_1, z_2, ..., z_T\}$ as the bridge to connect the different types of objects (authors, papers, and publication venues). More accurately, for each object it estimates a mixture of topic distribution which represents the probability of the object being associated with every topic. For example, for each author, we have a set of probabilities $\{p(z_i|a)\}$ and for each paper d, we have probabilities $\{p(z_i|d)\}$. For a given query q, we can use the obtained topic model to do inference and obtain a set of probabilities $\{p(z_i|q)\}$. Table 1 gives an example of the most relevant topics for the query "Database".

3.3 *DivLearn*: Learning to Diversify Expert Finding with Subtopics

Objective Function. Without considering diversification, we can use any learning-to-rank methods [11] to learn a model for ranking experts. For example, given a training data set (e.g., users' click-through data), we could maximize normalized discounted cumulative gain (NDCG) or Mean Average Precision (MAP). In this section, we use MAP as the example in our explanation. Basically, MAP is defined as:

Table 1. Most relevant topics (i.e., with a higher $p(z|q)$) for query "Database"

Topic	$p(z\|q)$
Topic 127: Database systems	0.15
Topic 134: Gene database	0.09
Topic 106: Web database	0.07
Topic 99: XML data	0.05
Topic 192: Query processing	0.04

$$MAP(Q,k) = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{\sum_{i=1}^{k} Prec(a_{ji}) \times rel(a_{ji})}{\sum_{i=1}^{k} rel(a_{ji})} \tag{2}$$

where Q is a set of queries in the training data; $Prec(a_{ji})$ represents the precision value obtained for the set of top i returned experts for query q_j; $rel(a_{ji})$ is an indicator function equaling 1 if the expert a_{ji} is relevant to query q_j, 0 otherwise. The normalized inner sum denotes the average precision for the set of top k experts and the normalized outer sum denotes the average over all queries Q.

Now, we redefine the objective function based on a generalized MAP metric called MAP-Topic, which explicitly incorporates the diversity of subtopics. More specifically, given a training data set $\{(q^{(j)}, A_{q^{(j)}})\}$, where $q^{(j)} \in Q$ is query and $A_{q^{(j)}}$ is the set of related experts for query $q^{(j)}$, we can define the following objective function:

$$\mathcal{O} = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\sum_{z=1}^{T} p(z|q) \times \frac{\sum_{i=1}^{k} rel(a_{ji}) \times \frac{\sum_{m=1}^{i} p(z|a_{jm})}{i}}{\sum_{i=1}^{k} p(z|a_{ji}) \times rel(a_{ji})} \tag{3}$$

where $rel(a_{ji})$ is an indicator function with a value of 1 if a_{ji} is in $A_{q^{(j)}}$, 0 otherwise.

Linear Ranking Model. To instantiate the expert ranking model, we define different features. For example, for expert finding in the academic network, we define features such as the number of publications, h-index score of the author, and the language model-based relevance score. For the i-th feature, we define $\phi_i(a,q)$ as the feature value of author a to the given query q. Finally, without loss of generality, we consider the linear model to calculate the score for ranking experts, thus have

$$s(a,q) = w^T \Phi\,(a,q) = \sum_{i=1}^{N} w_i \phi_i\,(a,q) \tag{4}$$

where w_i is the weight of the i-the feature. Given a feature weight vector w, according to the objective function described above, we can calculate a value, denoted as $\mathcal{O}(w)$, to evaluate the ranking results of that model. Thus our target is to find a configuration of w to maximize $\mathcal{O}(w)$.

3.4 Model Learning

Many algorithms can be used for finding the optimal w in our model, such as hill climbing [15], gene programming(GP) [10], random walk, gradient descent [4]. For the

Algorithm 1. Model learning algorithm.

```
Input: training samples $S$
Output: learned parameters $w$
Initialize $globalBestW, step$
for $circle = 1 \rightarrow loop$ do
    $w \leftarrow empiricalVector + randomVector$
    repeat
        $w_{new} \leftarrow w + step$
        if $\mathcal{O}(w_{new}) > \mathcal{O}(w)$ then
            $w \leftarrow w_{new}$
        else
            Update $step$
        end if
    until convergence
    if $\mathcal{O}(w) > \mathcal{O}(globalBestW)$ then
        $globalBestW \leftarrow w$
    end if
end for
return $globalBestW$
```

purpose of simplicity and effectiveness, in this paper, we utilize the hill climbing algorithm due to its efficiency and ease of implementation. The algorithm is summarized in Algorithm 1.

Different from the original random start hill climbing algorithm which starts from pure random parameters, we add our prior knowledge $empiricalVector$ to the initialization of w, as we know some features such as BM25 will directly affect the relevance degree tends to be more important. By doing so, we could reduce the CPU time for training.

4 Experiment

We evaluate the proposed models in an online system, Arnetminer[1].

4.1 Experiment Setup

Data Sets. From the system, we obtain a network consisting of 1,003,487 authors, 6,687 conferences, and 2,032,845 papers. A detailed introduction about how the academic network has been constructed can be referred to [19]. As there is no standard data sets available, and also it is difficult to create such an data sets with ground truth. For a fair evaluation, we construct a data set in the following way: First, we select a number of most frequent queries from the query log of the online system; then we remove the overly specific or lengthy queries (e.g., ‘A Convergent Solution to Subspace Learning’) and normalize similar queries (e.g., ‘Web Service’ and ‘Web Services’ to ‘Web

[1] http://arnetminer.org

Table 2. Statistics of selected queries. $Entropy(q) = -\sum_{i=1}^{T} p(z_i|q) \log p(z_i|q)$ measures the query's uncertainty; $\#(\tau = 0.2)$ denotes the minimum number of topics that satisfy $\sum P(z|q) \geq \tau$.

Query	Data Source Venue	Entropy	$\#(\tau = 0.1)$	$\#(\tau = 0.2)$
Data Mining	KDD 08-11	4.9	3	5
Information Retrieval	SIGIR 08-11	4.8	3	8
Software Engineering	ICSE 08-11	4.5	1	3
Machine Learning	NIPS 08-11 & ICML 08-11	4.6	2	4

Service'). Second, for each query, we identify the most relevant (top) conferences. For example, for 'Web Service', we select ICWS and for 'Information Retrieval', we select SIGIR. Then, we collect and merge PC co-chairs, area chairs, and committee members of the identified top conferences in the past four years. In this way, we obtain a list of candidates. We rank these candidates according to the appearing times, breaking ties using the h-index value [8]. Finally, we use the top ranked 100 experts as the ground truth for each query.

Topic Model Estimation. For the topic model (ACT), we perform model estimation by setting the topic number as 200, i.e., $T = 200$. The topic number is determined by empirical experiments (more accurately, by minimizing the perplexity [2], a standard measure for estimating the performance of a probabilistic model, the lower the better). The topic modeling is carried out on a server running Windows 2003 with Dual-Core Intel Xeon processors (3.0 GHz) and 4GB memory. For the academic data set, it took about three hours to estimate the ACT model.

We produce some statistics for the selected queries (as shown in Table 2). $Entropy(q)$ measures the query's uncertainty and $\#(\tau = 0.2)$ denotes the minimum number of topics that satisfy $\sum P(z|q) \geq \tau$.

Feature Definition. We define features to capture the observed information for ranking experts of a given query. We consider two types of features: 1) query-independent features (such as h-index, sociability, and longevity) and 2) query-dependent features (such as BM25 [13] score and language model with recency score). A detailed description of the feature definition is given in Appendix.

Evaluation Measures and Comparison Methods. To quantitatively evaluate the proposed method, we consider two aspects: relevance and diversity. For the feature-based ranking, we consider six-fold cross-validation(i.e. five folds for training and the rest for testing) and evaluate the approaches in terms of Prec@5, Prec@10, Prec@15, Prec@20, and MAP. And we conduct evaluation on the entire data of the online system (including 916,946 authors, 1,558,499 papers, and 4,501 conferences). We refer to the proposed method as *DivLearn* and compare with the following methods:

RelLearn: A learning-based method. It uses the same setting (the same feature definition and the same training/test data) as that in *DivLearn*, except that it does not consider the topic diversity and directly use MAP as the objective function for learning.

Table 3. Performance for expert search approaches (%)

Approach	Prec@5	Prec@10	Prec@15	Prec@20	MAP
LM	21.1	18.3	15.6	13.6	26.3
BM25	17.8	16.7	14.8	14.7	27.1
ACT	23.3	22.8	21.5	21.4	32.3
ACT+RW	21.1	20.6	20.7	18.9	34.6
LDA	12.2	15	15.9	15.6	20.4
pLSI	21.1	21.1	19.3	18.6	31.8
RelLearn	27.8	24.4	24.8	**26.1**	35.8
DivLearn	**35.6**	**28.3**	**26.7**	25.8	**41.3**

Language Model: Language model(LM) [2] is one of the state-of-the-art approaches for information retrieval. It defines the relevance between an expert (document) and a query as a generative probability: $p(q|d) = \prod_{w \in q} p(w|d)$.

BM25 [13]: Another state-of-the-art probabilistic retrieval model for information retrieval.

pLSI: Hofmann proposes the probabilistic Latent Semantic Indexing(pLSI) model in [9]. After modeling, the probability of generating a word w from a document d can be calculated using the topic layer: $p(w|d) = \sum_{z=1}^{T} p(w|z)p(z|d)$. To learn the model, we use the EM algorithm[9].

LDA: Latent Dirichlet Allocation (LDA) [5] also models documents by using a topic layer. We performed model estimation with the same setting as that for the ACT model.

ACT: ACT model is presented in §3. As the learned topics is usually general and not specific to a given query, only using it alone for modeling is too coarse for academic search [22], so the final relevance score is defined as a combination with the language model $p(q|a) = p_{ACT}(q|a) \times p_{LM}(q|a)$.

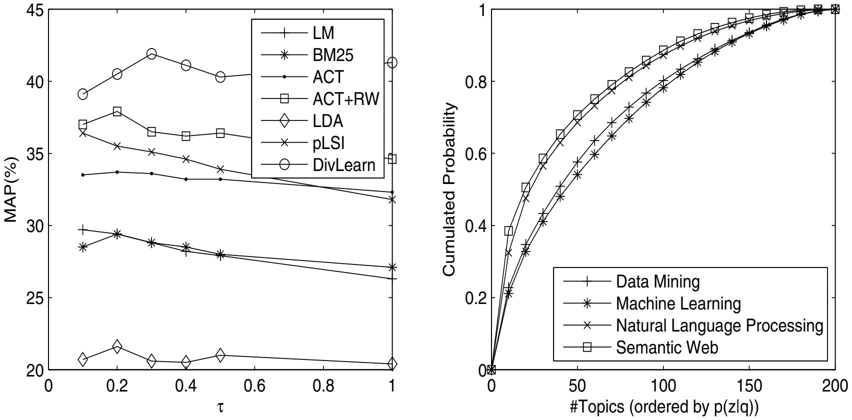
ACT+RW: A uniform academic search framework proposed in [17], which combines random walk and the ACT model together.

4.2 Performance Comparison

Table 3 lists the performance results of the different comparison methods. It can be clearly seen that our learning approach significantly outperforms the seven comparison methods. In terms of P@5, our approach achieves a +23% improvement compared with the (LDA). Comparing with the other expert finding methods, our method also results in an improvement of 8-18%. This advantage is due to that our method could combine multiple sources of evidences together. From Table 3, we can also see that the learning-based methods (both RelLearn and DivLearn) outperform the other relevance-based methods in terms of all measurements. Our DivLearn considers the diversity of topics, thus further improve the performance.

4.3 Analysis and Discussion

Now, we perform several analysis to examine the following aspects of *DivLearn*:(1) convergence property of the learning algorithm; (2) effect of different topic threshold; and (3) effect of recency impact function in Eq. 7.



(a) Effect of threshold τ on MAP

(b) Cumulated probabilities $\sum_z p(z|q)$ for top topics of example queries

Fig. 2. Effect of topic threshold analysis

Convergence Property. We first study the convergence property of the learning algorithm. We trace the execution of 72 random hill climbing runs to evaluate the convergence of the model learning algorithm. On average, the number of iterations to find the optimal parameter w varies from 16 to 28. The CPU time required to perform each iteration is around 1 minute. This suggests that the learning algorithm is efficient and has a good convergence property.

Effect of Topic Threshold. We conduct an experiment to see the effect of using different thresholds τ to select topics in the objective function (Eq. 3). We select the minimum number of topics with higher probabilities that statisfy $\sum_z p(z|q) \geq \tau$, then re-scale this sum to be 1 and assign 0 to other topics. Clearly, when $\tau = 1$, all topics are counted. Figure 2a shows the value of MAP of multiple methods for various τ. It shows that this metrics is consistent to a certain degree. The performance of different methods are relatively stable with different parameter setting. This could be explained by Figure 2b, which depicts the cumulated $P(z|q)$ of top n topics. As showed, for a given query, $p(z|q)$ tends to be dominated by several top related topics. Statistics in Table 2 also confirm this observation. All these observations confirm the effectiveness of the proposed method.

Effect of Recency. We evaluate whether expert finding is dynamic over time. In Eq. 7, we define a combination feature of the language model score and the recency score (Func 1). Now, we qualitatively examine how different settings for the recency impact function will affect the performance of *DivLearn*. We also compared with some other recency function with $Recency(p) = 2^{\left(\frac{\text{d.year - current year}}{\lambda}\right)}$ (Func 2) [14]. Figure 3 shows the performance of MAP with different parameter λ. The baseline denote the performance without considering recency. It shows that recency is an important factor and both impact functions perform better than the baseline which does not consider the recency.

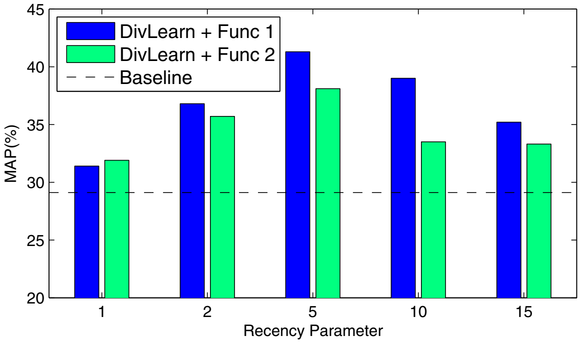


Fig. 3. MAP for different recency impact functions with different parameters

We can also see that both impact function perform best with the setting of $\lambda \simeq 5$. On average, the first impact function (Func 1, used in our approach) performs a bit better than Func 2.

5 Related Work

Previous works related to our learning to diversify for expert finding with subtopics can be divided into the following three aspects: expert finding, learning to rank, search result diversification. On expert finding, [17] propose a topic level approach over heterogenous network. [3] extended language models to address the expert finding problem. TREC also provides a platform for researchers to evaluate their models[16]. [7] present a learning framework for expert finding, but only relevance is considered. Other topic model based approaches were proposed either[17].

Learning to rank aims to combining multiple sources of evidences for ranking. Liu [11] gives a survey on this topic. He categorizes the related algorithms into three groups, namely point-wise, pair-wise and list-wise. To optimize the learning target, in this paper we use an list-wise approach, which is similar to [23].

Recently, a number of works study the problem of result diversification by taking inter-document dependencies into consideration [1,25,6,18]. Yue and Joachims [24] present a SVM-based approach for learning a good diversity retrieval function. For evaluation, Agrawal et al. [1] generalize classical information retrieval metrics to explicitly account for the value of diversification. Zhai et al. [25] propose a framework for evaluating retrieval different subtopics of a query topic. However, no previous work has been conducted for learning to diversify expert finding.

6 Conclusion

In this paper, we study the problem of learning to diversify expert finding results using subtopics. We formally define the problem in a supervised learning framework. An objective function is defined by explicitly incorporating topic-based diversity into the relevance based ranking model. An efficient algorithm is presented to solve the objective function. Experiment results on a real system validate the effectiveness of the proposed approach.

Learning to diversify expert finding represents a new research direction in both information retrieval and data mining. As future work, it is interesting to study how to incorporate diversity of relationships between experts into the learning process. In addition, it would be also interesting to detect user intention and to learn weights of subtopics via interactions with users.

References

1. Agrawal, R., Gollapudi, S., Halverson, A., Ieong, S.: Diversifying search results. In: WSDM 2009, pp. 5–14. ACM (2009)
2. Baeza-Yates, R., Ribeiro-Neto, B., et al.: Modern information retrieval, vol. 463. ACM Press, New York (1999)
3. Balog, K., Azzopardi, L., de Rijke, M.: Formal models for expert finding in enterprise corpora. In: SIGIR 2006, pp. 43–50. ACM (2006)
4. Bertsekas, D.: Nonlinear programming. Athena Scientific, Belmont (1999)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. In: NIPS 2001, pp. 601–608 (2001)
6. Carbonell, J.G., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: SIGIR 1998, pp. 335–336 (1998)
7. Fang, Y., Si, L., Mathur, A.: Ranking experts with discriminative probabilistic models. In: SIGIR 2009 Workshop on LRIR. Citeseer (2009)
8. 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)
9. Hofmann, T.: Probabilistic latent semantic indexing. In: SIGIR 1999, pp. 50–57. ACM (1999)
10. Koza, J.: On the programming of computers by means of natural selection, vol. 1. MIT Press (1996)
11. Liu, T.: Learning to rank for information retrieval. Foundations and Trends in Information Retrieval 3(3), 225–331 (2009)
12. Radlinski, F., Bennett, P.N., Carterette, B., Joachims, T.: Redundancy, diversity and interdependent document relevance. SIGIR Forum 43, 46–52 (2009)
13. Robertson, S., Walker, S., Hancock-Beaulieu, M., Gatford, M., Payne, A.: Okapi at trec-4. In: Proceedings of TREC, vol. 4 (1995)
14. Roth, M., Ben-David, A., Deutscher, D., Flysher, G., Horn, I., Leichtberg, A., Leiser, N., Matias, Y., Merom, R.: Suggesting friends using the implicit social graph. In: KDD 2010 (2010)
15. Russell, S., Norvig, P., Canny, J., Malik, J., Edwards, D.: Artificial intelligence: a modern approach, vol. 74. Prentice Hall, Englewood Cliffs (1995)
16. Soboroff, I., de Vries, A., Craswell, N.: Overview of the trec 2006 enterprise track. In: Proceedings of TREC. Citeseer (2006)
17. Tang, J., Jin, R., Zhang, J.: A topic modeling approach and its integration into the random walk framework for academic search. In: ICDM 2008, pp. 1055–1060 (2008)
18. Tang, J., Wu, S., Gao, B., Wan, Y.: Topic-level social network search. In: KDD 2011, pp. 769–772. ACM (2011)
19. Tang, J., Yao, L., Zhang, D., Zhang, J.: A combination approach to web user profiling. ACM TKDD 5(1), 1–44 (2010)

20. 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)
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. Wei, X., Croft, W.: Lda-based document models for ad-hoc retrieval. In: SIGIR 2006, pp. 178–185. ACM (2006)
23. Yeh, J., Lin, J., Ke, H., Yang, W.: Learning to rank for information retrieval using genetic programming. In: SIGIR 2007 Workshop on LR4IR. Citeseer (2007)
24. Yue, Y., Joachims, T.: Predicting diverse subsets using structural svms. In: ICML 2008, pp. 1224–1231. ACM (2008)
25. Zhai, C., Cohen, W., Lafferty, J.: Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In: SIGIR 2003, pp. 10–17. ACM (2003)

Appendix: Feature Definition

This section depicts how we define features in our experiment. In total, we defined features of two categories: *query-independent* and *query-dependent*.

- *h-index:* h-index equals h indicates that an author has h of N papers with at least h citations each, while the left $(N - h)$ papers have at most h citations each.
- *Longevity:* Longevity reflects the length of an author's academic life. We consider the year when one author published his/her first paper as the beginning of his/her academic life and the last paper as the end year.
- *Sociability:* The score of an author's sociability is defined based on how many coauthor he/she has. This score is defined as:

$$Sociability(A) = 1 + \sum_{c \in \text{A's coauthors}} ln(\#co - paper_c) \tag{5}$$

  where $\#co - paper_c$ denotes the number of papers coauthored between the author and the coauthor c.
- *Language Model with Recency:* We consider the effect of recency and impact factor of conference. Thus the language model score we used for an author is redefined as:

$$LM(q|a) = \sum_{d \in \{\text{a's publications}\}} p(q|d) \times Impact(d.conference) \times Recency(d) \tag{6}$$

  where $Recency(d)$ for publication d is defined as:

$$Recency(d) = \exp\left(\frac{\text{d.year - current year}}{\lambda}\right) \tag{7}$$

- *BM25 with Recency:* It defines a similar relevance score as that in Eq. 6, except that the $p(q|d)$ is obtained by BM25.

An Associative Classifier for Uncertain Datasets

Metanat Hooshsadat and Osmar R. Zaïane

University of Alberta
Edmonton, Alberta, Canada
{hooshsad,zaiane}@cs.ualberta.ca

Abstract. The classification of uncertain datasets is an emerging research problem that has recently attracted significant attention. Some attempts to devise a classification model with uncertain training data have been proposed using decision trees, neural networks, or other approaches. Among those, the associative classifiers have inspired some of the uncertain classification algorithms given their promising results on standard datasets. We propose a novel associative classifier for uncertain data. Our method, Uncertain Associative Classifier (UAC) is efficient and has an effective rule pruning strategy. Our experimental results on real datasets show that in most cases, UAC reaches better accuracies than the state of the art algorithms.

1 Introduction

Typical relational databases or databases in general hold collections of records representing facts. These facts are observations with known values stored in the fields of each tuple of the database. In other words, the observation represented by a record is assumed to have taken place and the attribute values are assumed to be true. We call these databases "certain database" because we are certain about the recorded data and their values. In contrast to "certain" data there is also "uncertain data"; data for which we may not be sure about the observation whether it really took place or not, or data for which the attribute values are not ascertained with 100% probability.

Querying such data, particularly computing aggregations, ranking or discovering patterns in probabilistic data is a challenging feat. Many researchers have focused on uncertain databases, also called probabilistic databases, for managing uncertain data [1], top-k ranking uncertain data [2], querying uncertain data [3], or mining uncertain data [4,5]. While many approches use an existancial uncertainty attached to a record as a whole, our model targets uncertain databases with probabilities attached to each attribute value.

This paper addresses the problem of devising an accurate rule-based classifier on uncertain training data. There are many classification paradigms but the classifiers of interest to our study are rule-based. We opted for associative classifiers, classifiers using a model based on association rules, as they were shown to be highly accurate and competitive with other approaches [6].

After briefly reviewing related work for associative classification as well as published work on classifying in the presence of uncertainty, we present in Section

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 342–353, 2012.

3 our novel classification method UAC. Finally in Section 4 we present empirical evaluations comparing UAC with other published works.

2 Related Works

Recently, a considerable amount of studies in machine learning are directed toward the uncertain data classification, including: TSVC [7] (inspired by SVM), DTU [8] (decision tree), UNN [9] (based on Neural Network), a Bayesian classifier [10], uRule [11] (rule based), uHARMONY [12] and UCBA [13] (based on associative classifiers). However, models suggested by the previous work do not capture some possible types of uncertainty. In previous studies, numerical attributes are only modeled by intervals, while they may exist in other forms such as probability vectors. Categorical attributes are modeled by a probability distribution vector over their domain where the vector is unrealistically assumed to be completely known. We use a probability on each attribute value.

High accuracy and strong flexibility are some of the advantageous characteristics of the rule based classifiers. Investigating rule based uncertain data classifiers has been the theme of many studies. One of these studies is uRule [11], which defines the information gain metric in presence of uncertainty. The probability of each rule classifying the instance is computed based on the weighting system introduced by uRule.

Associative classification is a large category of rule based classification in which the rule induction procedure is based on the association rule mining technique. Some of the prominent associative classifiers are CBA [14], ARC [15], and CMAR [16]. In this paper, we introduce an associative classifier for uncertain datasets, which is based on CBA. CBA is highly accurate, flexible and efficient both in time and memory [14].

CBA directly adopts Apriori to mine the potential classification rules or strong *ruleitems* from the data. Ruleitems are those association rules of form $a \rightarrow c$, where the consequence (c) is a class label and the antecedent (a) is a set of *attribute assignments*. Each *attribute assignment* consists of an attribute and a value which belongs to the domain of that attribute. For example, if A_1 and A_2 are two attributes and c is a class label, $r = (A_1 : u_1, A_2 : u_2 \rightarrow c)$ is a ruleitem. r implies that if A_1 and A_2 have values of u_1 and u_2 respectively, the class label should be c. A ruleitem is strong if its support and confidence are above the predefined thresholds.

After mining the strong ruleitems, a large number of them are eliminated by applying the *database coverage* approach. This method of filtering rules is applied by all rule-based classifiers, particularly associative classifiers. However, in the case of uncertain data, database coverage presents a significant challenge. Rule based classifiers often need to evaluate various rules to pick the best ones. This level is critical in maintaining a high accuracy. The evaluation often involves the answer to the following question: *To which training instances can a rule be applied?* Yet, the answer is not obvious for uncertain datasets. Many uncertain dataset instances may satisfy the antecedent of a rule, each with a different probability. Existing uncertain data rule based classifiers have suggested various answers to this problem.

uHARMONY suggested a lower bound on the probability by which the instance satisfies the rule antecedent. This approach is simple and fast, but the difficulty or even impossibility of setting the threshold is a problem. This is explained in more detail in Section 3.2. uRule suggested to remove the items in the antecedent of the rule from the instance, to leave only the uncovered part of the instance every time. In contrast to uHARMONY, this method uses the whole dataset but it may cause sensitivity to noise which is undesirable. UCBA, wich is based on CBA, does not include the uncertainty in the rule selection process; they select as many rules as possible. This method does not filter enough rules; so may decrease the accuracy.

In UAC, we introduce a new solution to the coverage problem. This computation does not increase the running time complexity and needs no extra passes over the dataset.

3 UAC Algorithm

In this section, we present our novel algorithm, *UAC*. Before applying UAC to uncertain numerical attributes in the train sets, they are first transformed into uncertain categorical attributes using U-CAIM [10], assuming the normal distribution on the intervals. After discretization, the value of the i-th attribute for the j-th instance is a list of value-probability pairs, as shown in Equation 1.

$$\begin{array}{l} A_{j,i} = \{(x_{j,i,1} : p_{j,i,1}), (x_{j,i,2} : p_{j,i,2}), .., (x_{j,i,k} : p_{j,i,k})\} \\ \forall q \leq k \, ; A_j.l \leq x_{j,i,q} \leq A_j.u \, \Sigma_{q=1}^{k} p_{j,i,q} = 1. \end{array} \tag{1}$$

Building an associative classifier consists of two distinct steps: 1- Rule Extraction, 2- Rule Filtering. In this section each step of UAC is explained. Later, the procedure of classifying a new test instance is described.

3.1 Rule Extraction

In uncertain datasets, an association rule is considered strong if it is frequent and its *confidence (Conf)* is above a user defined threshold called *minimum confidence*. A ruleitem is frequent if its *Expected Support (ES)* is above a user defined threshold called *minimum expected support*. The definitions of the expected support and the confidence are as follows.

Definition. If a is an itemset and c is a class label, expected support (ES) and confidence (Conf) of a ruleitem are calculated by Equation 2. Here, the ruleitem is denoted by $r = a \to c$ and T is the set of all transactions.

$$\begin{array}{ll} ES(a) & = \Sigma_{\forall t \in T} \Pi_{\forall i \in a} P(i \in t) \\ ES(a \to c) & = \Sigma_{\forall t \in T, t.class=c} \Pi_{\forall i \in a} P(i \in t). \\ Conf(a \to c) & = \frac{ES(a \to c)}{ES(a)}. \end{array} \tag{2}$$

Some studies have criticized expected support and defined another measure which is called probabilistic support [17] [18]. Probabilistic support is defined as the probability of an itemset to be frequent with respect to a certain minimum expected support. However, probabilistic support increases the time complexity significantly. Therefore to be more efficient, UAC uses the expected support.

uHARMONY defines another measure instead of confidence which is called *expected confidence*. The computation of this measure takes $O(|T|^2)$ time where $|T|$ is the number of instances. Computing confidence is only $O(1)$, thus we use confidence for efficiency reasons. Our experimental results in Section 4 empirically shows that our confidence based method can reach high accuracies.

Our rule extraction method is based on UApriori [4]. The candidate set is first initialized by all rules of form $a \rightarrow c$ where a is a single *attribute assignment* and c is a class label. After removing all infrequent ruleitems, the set of candidates is pruned by the pessimistic error rate method [19]. Each two frequent ruleitems with the same class label are then joined together to form the next level candidate set. The procedure is repeated until the generated candidate set is empty, meaning all the frequent ruleitems have been found. Those ruleitems that are strong (their confidence is above the predefined threshold) are the potential classification rules. In the next section, the potential ruleitems are filtered and the final set of rules is formed.

3.2 Rule Filtering

The outcome of the rule extraction is a set of rules called *rawSet*. Usually the number of ruleitems in rawSet is excessive. Excessive rules may have negative impact on the accuracy of the classification model. To prevent this, UAC uses the database coverage method to reduce the set of rules while handling the uncertainty. The initial step of the database coverage method in UAC is to sort rules based on their absolute precedence to accelerate the algorithm. Absolute precedence in the context of uncertain data is defined as follows:

Definition: Rule r_i has absolute precedence over rule r_j or $r_i \succ r_j$, if *a)* r_i has higher confidence than r_j; *b)* r_i and r_j have the same confidence but r_i has higher expected support than r_j; *c)* r_i and r_j have the same confidence and the same expected support but r_i have less items in its antecedent than r_j.

When data is not uncertain, confidence is a good and sufficient measure to examine whether a rule is the best classifier for an instance. But when uncertainty is present, there is an additional parameter in effect. To illustrate this issue, assume rules $r_1 : [m, t \rightarrow c_1]$ and $r_2 : [n \rightarrow c_2]$ having confidences of 0.8 and 0.7, respectively. It is evident that $r_1 \succ r_2$. However, for a test instance like $I_1 : [(m : 0.4), (n : 0.6), (t, 0.3) \rightarrow x]$ where x is to be predicted, which rule should be used? According to CBA, r_1 should be used because its confidence is higher than that of r_2. However, the probability that I_1 satisfies the antecedent of r_1 is small, so r_1 is not likely to be the right classifier. We solve this problem by including another measure called *PI*. *PI* or *probability of inclusion*, denoted by $\pi(r_i, I_k)$, is described as the probability by which rule r_i can classify instance I_k. *PI* is defined in Equation 3. In the example above $\pi(r_1, I_1)$ is only $0.3 \times 0.4 = 0.12$, While $\pi(r_2, I_1)$ is 0.6.

$$\pi(r_i, I_k) = \Pi_{w \in r_i} P(w \in I_k). \tag{3}$$

Next, we define *applicability*, denoted by $\alpha(r_i, I_k)$ in Equation 4. Applicability is the probability by which rule r_i correctly classifies instance I_k and is used as one of the main metrics in UAC. For the previous example, $\alpha(r_1, I_1) = 0.096$ and $\alpha(r_2, I_1) = 0.42$. Thus, it is more probable that I_1 is correctly classified by r_2 than r_1.

$$\alpha(r_i, I_k) = r_i.Conf \times \pi(r_i, I_k). \tag{4}$$

Now based on the applicability, we define the concept of relative precedence of rule r_i over rule r_j with respect to I_k. This is denoted by $r_i \succ_{[I_k]} r_j$ and is defined as follows:

Definition: Rule r_i has relative precedence over rule r_j with respect to instance I_k denoted by $r_i \succ_{[I_k]} r_j$, if: *a)* $\alpha(r_i, I_k) > \alpha(r_j, I_k)$ *b)* r_i and r_j have the same applicability with respect to I_k but r_i has absolute precedence over r_j. Having $r_i \succ_{[I_k]} r_j$ implies that r_i is "more reliable" than r_j in classifying I_k. It is evident from the definition, that the concept of "more reliable" rule in an uncertain data classifier is relative. One rule can be more reliable than the other when dealing with an instance, and the opposite may be true for another instance. In the previous example, r_2 has relative precedence over r_1, even though r_1 has absolute precedence over r_2.

UAC uses the relative precedence as well as the absolute precedence to filter rawSet. The database coverage algorithm of UAC has 3 stages that are explained below.

Stage 1: Finding ucRules and uwRules. After sorting rawSet based on the absolute precedence, we make one pass over the dataset to link each instance i in the dataset to two rules in rawSet: *ucRule* and *uwRule*. *ucRule* is the rule with the highest relative precedence that correctly classifies i. In contrast, *uwRule* is the rule with the highest relative precedence that wrongly classifies i. The pseudocode for the first stage is presented in Algorithm 1.

In Algorithm 1, three sets are declared. U contains all the rules that classify at least one training instance correctly. Q is the set of all *ucRule*s which have relative precedence over their corresponding *uwRule*s with respect to the associated instances. If *i.uwRule* has relative and absolute precedence over the corresponding *ucRule*, a record of form $< i.id, i.class, ucRule, uwRule >$ is put in A. Here, *i.id* is the unique identifier of the instance and *i.class* represents the class label.

To find the corresponding *ucRule* and *uwRule* for each instance, the procedure starts at the first rule of the sorted rawSet and descends. For example, if there is a rule that correctly classifies the target instance and has applicability of α, we pass this rule and look for the rules with higher applicabilities to assign as *ucRule*. Searching continues only until we reach a rule that has a confidence of less than α. Clearly, this rule and rules after it (with less confidence) have no chance of being *ucRule*. The same applies to *uwRule*. Also as shown in Algorithm 1 lines 4 and 6, the applicability values of *ucRule* and *uwRule* are stored to expedite the process for the next stages.

The purpose of the database coverage in UAC is to find the best classifying rule (coverage) for each instance in the dataset. The covering rules are then contained in the final set of rules and others are filtered out. The best rule, that is the covering rule, in CBA is the highest precedence rule that classifies an instance. This definition is not sufficient for UAC because the highest precedence rule may have a small PI.

To solve the aforementioned problem, uHARMONY sets a predefined lower bound on the PI value of the covering rule, a method with various disadvantages. Clearly, not only estimating the suitable lower bound is critical, but it is also intricate, and even in many cases impossible. When predicting a label for an instance, rules that have higher PI than the lower bound are treated alike. To improve upon this, it is necessary to set the lower bound high enough to avoid low probability rules covering the instances. However, it remains that it is possible that the only classifying rules for some of the instances are not above that lower bound and are removed. Additionally, setting a predefined lower bound filters out usable information, while the purpose of the uncertain data classifiers is to use all of the available information. Moreover, having a single bound for all of the cases is not desirable. Different instances may need different lower bounds.

Given all the above reasons, we need to evaluate the suitable lower bound for each instance. The definition of the covering rule in UAC is as follows, where we use the applicability of *i.ucRule* as our lower bound for covering i.

Definition: Rule r covers instance i if: *a*) r classifies at least one instance correctly; *b*) $\pi(r,i) > 0$; *c*) $\alpha(r,i) > \alpha(i.ucRule, i) = cApplic$. *d*) $r \succ i.ucRule$

cApplic represents the maximum rule applicability to classify an instance correctly. Thus, it is the suitable lower bound for the applicability of the covering rules. This will ensure that each instance is covered with the best classifying rule (*ucRule*) or a rule with higher relative and absolute precedence than *ucRule*. In the next two stages, we remove the rules that do not cover any instance from rawSet.

Stage 2: Managing Replacements. In this stage (Algorithm 2), cases that were stored in A at Stage 1 are managed. A contains all cases where *i.uwRule* has relative and absolute precedence over *i.ucRule*, thus *i.ucRule* may not cover i. If *i.uwRule* is flagged in Stage 1, i is covered by *i.uwRule* (lines 3, 4, and 5). Otherwise based on the definition of the covering rule in Stage 1, i may get the coverage by the other rules such as w which have the following characteristics: *a*) w classifies i incorrectly; *b*) w has relative precedence over *i.ucRule* with respect to i; *c*) w has absolute precedence over *i.ucRule*.

Function *allCoverRules* (line 7) finds all such rules as w within U, which are called the replacements of *i.ucRule*. The replacement relation is stored in a DAG (directed acyclic graph) called *RepDAG*. In RepDAG, each parent node has a

Algorithm 1. UAC Rule Filtering: Stage 1

```
1: $Q = \emptyset; U = \emptyset; A = \emptyset$
2: for all $i \in Dataset$ do
3:     $i.ucRule = firstCorrect(i)$
4:     $i.cApplic = \alpha(i.ucRule, i)$
5:     $i.uwRule = firstWrong(i)$
6:     $i.wApplic = \alpha(i.uwRule, i)$
7:     $U.add(ucRule)$
8:     $ucRule.covered[i.class] + +$
9:     if $(ucRule \succ_{[i]} uwRule)$ and $ucRule \succ uwRule$ then
10:        $Q.add(ucRule)$
11:        $flag(ucRule)$
12:    else
13:        $A.add(< i.id, i.class, ucRule, uwRule >)$
14:    end if
15: end for
```

pointer to each child node via the *replace* set (line 12). The number of incoming edges is stored in *incom* (line 14). Each node represents a rule and each edge represents a replacement relation.

Each rule has a *covered* array in UAC where *r.covered*[*c*] is used to store the total number of instances covered by r and labeled by class c. If $r.covered[r.class] = 0$, then r does not classify any training instance correctly and is filtered out. Starting from line 22, we traverse RepDAG in its topologically sorted order to update the *covered* array of each rule. Rule r_i comes before r_j in the sorted order, if $r_i \succ r_j$ and there is no instance such as I_k where $r_j \succ_{[I_k]} r_i$. If a rule fails to cover any instance correctly (line 26), it does not have any effect on the *covered* array of the rules in its replace set. At the end of this Stage, enough information has been gathered to start the next stage, which finalizes the set of rules.

Stage 3: Finalizing Rules. At stage 3 (Algorithm 3), the set of rules is finalized. In this Stage, UAC filters the rules based on a greedy method of error reduction. Function *computeError* counts the number of instances that are covered by rule r but have a different class label than *r.class*. The covered instances are then removed from the dataset. Function *addDefaultClass* finds the most frequent class label among the remaining instances (line 6). In line 8, the number of instances correctly classified by the default class is calculated. *totalError* is the total errors made by the current rule r and the default class. In fact, each rule with positive coverage over its class, is associated with a particular *totalError*, *defClass*, and *defAcc* (line 10). After processing the rules, we break the set of rules from the minimum error and assign *default* and *defApplic*. *defApplic* is used in rule selection as an estimate of applicability of the default class.

Our rule filtering algorithm has a runtime of $O(|T| \times |R|)$ in the worst case scenario, where $|T|$ is the number of instances in the dataset and $|R|$ is the size

Algorithm 2. UAC Rule Filtering: Stage 2

```
$RepDAG = \emptyset$
**for all** $< i.id, y, ucRule, uwRule > \in A$ **do**
  **if** $flagged(uwRule)$ **then**
    $ucRule.covered[y] - -$
    $uwRule.covered[y] + +$
  **else**
    $wSet = allCoverRules(U, i.id, ucRule)$
    **if** $!RepDAG.contains(ucRule)$ **then**
      $RepDAG.add(ucRule)$
    **end if**
    **for all** $w \in wSet$ **do**
      $w.replace.add(< ucRule, i.id, y >)$
      $w.covered + +$
      $ucRule.incom + +$
      **if** $!w \in RepDAG$ **then**
        $RepDAG.add(w)$
      **end if**
    **end for**
    $Q = Q.add(wSet)$
  **end if**
**end for**
$S \leftarrow$ set of all nodes with no incoming edges
**while** $S \neq \emptyset$ **do**
  $r = S.next()$ $\{$*next* removes a rule from the set$\}$
  **for all** $< ucRule, id, y > \in r.replace$ **do**
    **if** $(r.covered[r.class] > 0)$ **then**
      **if** *id* is covered **then**
        $r.covered[y] - -$
      **else**
        $ucRule.covered[y] - -$
        Mark *id* as covered.
      **end if**
    **end if**
    $ucRule.incom - -$
    **if** $ucRule.incom = 0$ **then**
      $S.add(ucRule)$
    **end if**
  **end for**
**end while**
```

of rawSet. The worst case scenario is when at Stage 1, at least one *ucRule* or *uwRule* is the last rule in the sorted rawSet. This case rarely happens because the rules are sorted based on their absolute precedence. UAC also makes slightly more than one pass over the dataset in the rule filtering step. Passes are made in Stage 1 and 2. Note that array A is usually small, given that most of the instances are usually classified by the highest ranked rules. The number of passes is an important point, because the dataset may be very large. Specially for datasets that can not be loaded into memory at once, it is not efficient to make multiple pases. This is an advantage for UAC over UCBA, which passes over the dataset once for each rule in rawSet. Next section explains the rule selection that is the procedure of classifying test instances based on the set of rules.

Algorithm 3. UAC Rule Filtering: Stage 3

```
 1: $C = \emptyset$
 2: for all $r \in Q$ do
 3:    if $r.covered[r.class] > 0$ then
 4:       $finalSet.add(r)$
 5:       $ruleErrors+ = computeError(r)$
 6:       $defClass = addDefaultClass()$
 7:       $defErrors = computeDefErr(defClass)$
 8:       $defAcc = addDefAcc(uncovered(D) - defErrors)$
 9:       $totalError = defErrors + ruleErrors$
10:       $C.add(r, totalError, defClass, defAcc)$
11:    end if
12: end for
13: Break C from the rule with minimum error
14: C contains the final set of rules
15: $default = defClass.get(C.size)$
16: $defApplic = \frac{defAcc.get(C.size)}{|T|}$
```

3.3 Rule Selection

Rule selection is the procedure of classifying a test instance. In the previous sections, excessive rules were filtered out from rawSet. The remaining set of rules is called finalSet and classifies the test instances. UAC selects one classifying rule for each instance. The selected classifying rule has the highest relative precedence with respect to the test instance.

The role of the default class (*default* in Algorithm 3 line 15) is to reduce the number of rules. The default class predicts the labels of those instances that are not classified by the rules in the finalSet. So the best predicting label for some of the test instances may be default class. But UAC may prefer rules with small PI values to the default class if we follow the procedure of "certain" data classifiers. To prevent this, *defApplic* is used as an estimate for applicability of the default rule. This value shows the number of training instances that were expected to be classified by the default rule. For example, when two classes, such as a and b, have the same population in the dataset but no rule labeled b exists, default rule has a very important role. Consequently, the value of the default applicability is high. As a result, if the highest precedence rule with respect to a test instance has less applicability than the default rule, the default rule will predict the label for that.

4 Experiments and Results

We use an empirical study to compare UAC against the existing rule based methods. In all of the reported experiments on UAC, the minimum support is set to 1%, the minimum confidence to 0.5 and the maximum number of mined association rules to $80,000$. Each reported number is an average over 10 repetitions of 10-fold cross validations.

Since there is no public repository of uncertain datasets, we synthetically added uncertainty to 28 well known UCI datasets. This method was employed by all the studies in the field including uHARMONY, DTU and uRule, uncertain svm, UCBA, etc. and gives a close estimation of the classifier performance in the real world problems. We selected the same datasets as in [12] to compare our method with the results reported in their paper for uHARMONY, uRule and DTU. This also ensures that we did not choose only the datasets on which our method performs better.

To compare our method against other classifiers, we employ averaging technique and case by case comparison [20]. The same method was employed by many other studies including CBA, uHARMONY, DTU and uRule to prove the better performance of their algorithms. Table 1 provides a comparison between UAC and other existing rule based methods in terms of accuracy. The reported accuracies for uHARMONY (#3), DTU (#4) and uRule (#5) are reproduced from [12]. We applied UAC (#2) to the same datasets generated by the same procedure of adding uncertainty as [12] to make the comparison meaningful. Value N/A, existing in the experiments reported by [12], shows that the classifier has run out of resources in their experiments. In Table 1, uncertainty level is U10@4 meaning that datasets have 10 percent uncertainty, where only four of the attributes with the highest information gain are uncertain. To add a level 10 uncertainty to an attribute, it is attached with a 0.9 probability and the remaining 0.1 is distributed randomly among the other values present in the domain. The accuracies in this table are reported on already discretized versions of the dataset that are available online and referenced in [12].

The accuracies reported show that in most cases UAC has reached higher accuracies. For some datasets the improvement is significantly high, such as *wine* dataset with 36.79% and *bands* dataset with 19.77% improvement over the existing maximum accuracy. UAC reaches higher accuracies on the average too.

We have conducted further extensive experiments comparing UAC and UCBA since both stem from CBA. Due to lack of space we report here only the summary and refer the reader to [21] for further details. Using a new and more general uncertainty model that we propose [21], we compared the accuracy of UAC and UCBA on all 28 datasets as in the previous experiments in Table 1 and show that UAC outperforms UCBA. On average, over the 28 datasets, the accuracy of UAC was 74.7% while UCBA averaged 67.5% if a sampled-based model is used when a numerical attribute is assigned a set of possible values; and respectively 70.3% versus 66.7% if an interval-based model is used [21]. In short, for a numerical attribute, the sampled-based model considers the attribute value to be expressed by a set of values with their respective probabilities, while the

Table 1. %Accuracy, reported by rule based classifiers on datasets modeled based on [12] at level of uncertainty of U4@10

#1 Dataset	#2 UAC	#3 uHAR	#4 DTU	#5 uRule
australian	80.2	**85.37**	83.62	84.35
balance	84.9	**89.3**	56.32	62.88
bands	**78.4**	58.63	N/A	N/A
breast	94.3	65.52	91.27	**94.56**
car	**89.3**	77.72	70.02	70.02
contracep	43.6	47.59	**50.1**	44.26
credit	78.1	**85.95**	84.35	74.35
echo	92	**93.29**	92.37	87.02
flag	45.7	52.42	**59.28**	44.85
german	71.9	69.6	**72.3**	70.1
heart	**77.3**	56.64	53.04	52.39
hepatitis	81.5	**82.52**	80	79.35
horse	72.4	82.88	**85.33**	N/A
monks-1	**99**	91.36	74.64	70.68
monks-2	**75.5**	65.72	65.72	65.72
monks-3	**98.1**	96.4	79.96	68.05
mushroom	**100**	97.45	**100**	99.98
pima	**73.8**	65.11	65.1	67.32
post_oper	58	69.75	**70**	**70**
promoters	66	69	**71.7**	61.32
spect	**81.8**	80.19	79.03	81.65
survival	**74**	73.53	73.53	72.55
ta_eval	**50.4**	45.04	48.34	33.77
tic-tac-toe	**90.8**	76.2	72.65	81.52
vehicle	**69.8**	63.44	64.78	N/A
voting	91.1	92.86	94.48	**94.94**
wine	**87.9**	51.11	42.13	41.57
zoo	**92.3**	88.76	92.08	89.11
Average	**78.5**	74.05	73.04	70.49

interval-based model considers the attribute value to be an interval with a probability distribution function. Moreover, comparing the training time, UAC was in many cases about 2 orders of magnitude faster (i.e. X100) than UCBA and produced significantly less rules for all tested uncertainty levels. This demonstrates the efficacy of the rule pruning startegy managing to preserve a better set of rules than UCBA.

5 Conclusion

In this paper we propose an effective way to prune associative classification rules in the presence of uncertainty and present a complete associative classifier for uncertain data that encompasses this pruning. Empirical results show that our algorithm outperforms 4 existing rule-based methods in terms of accuracy on average for 28 datasets and also show that UAC outperforms UCBA significantly for these 28 datasets in terms of accuracy even though UAC produces less classification rules and has a smaller runtime than UCBA.

References

1. Sen, P., Deshpande, A.: Representing and querying correlated tuples in probabilistic databases. In: IEEE ICDE, pp. 596–605 (2007)
2. Wang, C., Yuan, L.-Y., You, J.H., Zaiane, O.R., Pei, J.: On pruning for top-k ranking in uncertain databases. In: International Conference on Very Large Data Bases (VLDB), PVLDB, vol. 4(10) (2011)
3. Cheema, M.A., Lin, X., Wang, W., Zhang, W., Pei, J.: Probabilistic reverse nearest neighbor queries on uncertain data. IEEE Transactions on Knowledge and Data Engeneering (TKDE) 22, 550–564 (2010)
4. Aggarwal, C.C., Li, Y., Wang, J., Wang, J.: Frequent pattern mining with uncertain data. In: ACM SIGKDD, pp. 29–38 (2009)
5. Jiang, B., Pei, J.: Outlier detection on uncertain data: Objects, instances, and inference. In: IEEE ICDE (2011)
6. Antonie, M.-L., Zaiane, O.R., Holte, R.: Learning to use a learned model: A two-stage approach to classification. In: IEEE ICDM, pp. 33–42 (2006)
7. Bi, J., Zhang, T.: Support vector classification with input data uncertainty. In: Advances in Neural Information Processing Systems (NIPS), pp. 161–168 (2004)
8. 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)
9. Ge, J., Xia, Y., Nadungodage, C.: UNN: A Neural Network for Uncertain Data Classification. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010. LNCS, vol. 6118, pp. 449–460. Springer, Heidelberg (2010)
10. Qin, B., Xia, Y., Li, F.: A bayesian classifier for uncertain data. In: ACM Symposium on Applied Computing, pp. 1010–1014 (2010)
11. Qin, B., Xia, Y., Prabhakar, S., Tu, Y.: A rule-based classification algorithm for uncertain data. In: IEEE ICDE (2009)
12. Gao, C., Wang, J.: Direct mining of discriminative patterns for classifying uncertain data. In: ACM SIGKDD, pp. 861–870 (2010)
13. Qin, X., Zhang, Y., Li, X., Wang, Y.: Associative Classifier for Uncertain Data. In: Chen, L., Tang, C., Yang, J., Gao, Y. (eds.) WAIM 2010. LNCS, vol. 6184, pp. 692–703. Springer, Heidelberg (2010)
14. Liu, B., Hsu, W., Ma, Y.: Integrating Classification and Association Rule Mining. In: ACM SIGKDD, pp. 80–86 (1998)
15. Zaiane, O., Antonie, M.-L.: Classifying text documents by associating terms with text categories. In: Australasian Database Conference, pp. 215–222 (January 2002)
16. Li, W., Han, J., Pei, J.: CMAR: Accurate and efficient classification based on multiple class-association rules. In: IEEE ICDM, pp. 369–376 (2001)
17. Zhang, Q., Li, F., Yi, K.: Finding frequent items in probabilistic data. In: ACM SIGMOD, pp. 819–832 (2008)
18. Bernecker, T., Kriegel, H.P., Renz, M., Verhein, F., Zuefle, A.: Probabilistic frequent itemset mining in uncertain databases. In: ACM SIGKDD (2009)
19. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers (1993)
20. Demsar, J.: Statistical comparison of classifiers over multiple data sets. JMLR 7, 1–30 (2010)
21. Hooshsadat, M.: Classification and Sequential Pattern Mining From Uncertain Datasets. MSc dissertation, University of Alberta, Edmonton, Alberta (September 2011)

Neighborhood-Based Smoothing of External Cluster Validity Measures

Ken-ichi Fukui and Masayuki Numao

The Institute of Scientific and Industrial Research (ISIR),
Osaka University, 8-1 Mihogaoka, Ibaraki, Osaka, Japan
fukui@ai.sanken.osaka-u.ac.jp

Abstract. This paper proposes a methodology for introducing a neighborhood relation of clusters to the conventional cluster validity measures using external criteria, that is, class information. The extended measure evaluates the cluster validity together with connectivity of class distribution based on a neighborhood relation of clusters. A weighting function is introduced for smoothing the basic statistics to set-based measures and to pairwise-based measures. Our method can extend any cluster validity measure based on a set or pairwise of data points. In the experiment, we examined the neighbor component of the extended measure and revealed an appropriate neighborhood radius and some properties using synthetic and real-world data.

Keywords: cluster validity, neighborhood relation, weighting function.

1 Introduction

Clustering is a basic data mining task that discovers similar groups from given multi-variate data. Validation of a clustering result is a fundamental but difficult issue, since clustering is an unsupervised learning and is essentially to find latent clusters in the observed data[3,7,14]. Up until now, various validity measures have been proposed from different aspects, and they are mainly separated into two types whether based on internal or external criteria[7,10,8]:

- **Internal criteria** evaluate compactness and separability[3] of the clusters based only on distance between objects in the data space, that is *learning perspective*. As such measures, older methods of Dunn-index[4], DB-index[2], and recent CDbw[5] are well known. Surveys and comparisons of internal cluster validity measures are [3,9].
- **External criteria** evaluate how accurately the correct/desired clusters are formed in the clusters, that is *user's perspective*. External criteria normally uses class/category label together with cluster assignment. Purity, entropy, F-measure, and mutual information are typical measures[10,12,14].

This paper focuses on using external criteria, that is provided by human interpretation of data. It is more beneficial to use external criteria when class labels are available.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 354–365, 2012.

In order to understand obtained clusters better, this work introduces a neighborhood relation among clusters. A neighborhood relation is useful especially in case of micro-clusters or, i.e., cluster number is larger than class number. Global structure of clusters, which means not only individual (local) clusters, can be evaluated with neighborhood relation of classes within each cluster.

The basic policies of introducing the neighborhood relation is as follows:

1. A data object which belongs to the same class should be in neighbor over clusters. To evaluate this property, we introduce a weighting function based on *inter-cluster* distance. The inter-cluster distance can be computed based on either topology-based or Euclidean distance in the data space.
2. A weighting function is introduced into basic statistics that are commonly used in the conventional measures. Therefore, our approach is generic, any conventional cluster validity measure that uses these statistics can also be extended in the same way.

Above mentioned conventional indices do not consider neighboring clusters, while very few works introduce inter-cluster connectivity for prototype based clustering[11]. The inter-cluster connectivity is introduced by the first and the second best matching units, but this work is based on internal criterion. The contribution of this work is to introduce *neighborhood relation* over clusters into conventional external cluster validity indices.

The reason why we assume the situation to evaluate an unsupervised learning by class labels is as follows. The fundamental difficulty of unsupervised learning is that the features and the distance metric are derived from observation and assumption, there is no information from human interpretation of data. On the other hand, it is often the case that a small number of samples, or data from the same domain, or simulated samples are available with class labels. In such cases, an external validity measure works as a preliminary evaluation instead of evaluating unlabeled target data.

This paper presents how to introduce the weighting function to smooth the conventional clustering validity measures. In the experiment, we revealed the optimal smoothing radius and also examined several parameters, prototype (micro-cluster) number, and class overlapping degree. We revealed the properties of our extended measure and showed potential to validate a clustering result considering neighborhood relation of clusters.

2 Preliminaries

Definition 1. *(Clustering) Given a set of v-dimensional objects $\mathbf{S} = \{\boldsymbol{x}_i\}_{i=1}^{N} \in \mathbb{R}^v$, a clustering produces a cluster set $\mathbf{C} = \{C_i\}_{i=1}^{K}$ with a cluster assignment $c(i) \in \mathbf{C}$ for each object $\boldsymbol{x}_i$.*

Definition 2. *(Class) Let a class set be $\mathbf{T} = \{T_i\}_{i=1}^{L}$, and $t(i) \in \boldsymbol{T}$ denotes a class assignment for $\boldsymbol{x}_i$. Classes are provided independent from a clustering.*

Definition 3. *(Inter-cluster distance) $d(C_i, C_j) \in \mathbb{R}$ is defined as inter-cluster distance between clusters that can be computed either Euclidean-based or topology-based distance.*

Ex) Inter-cluster distance. Euclidean-based distances can be given by single linkage, complete linkage, and other methods commonly used in an aggregative hierarchical clustering. While, topology-based distance by the number of hops in a neighbor graph. The neighbor graph can be obtained by such as a threshold on Euclidean-based distance or by k-nearest neighbor.

Note that though a neighbor graph is normally obtained independent from a clustering process, some method produces cluster (vector quantization) with topology preservation such as Self-Organizing Map(SOM)[6], which is also used in this experiment.

The objective of this work is to evaluate density of class $\mathbf{T}$ within intra-cluster $\mathbf{C}$ together with the neighbor relation based on inter-cluster distance $d(C_i, C_j)$.

3 Neighborhood-Based Smoothing of Validity Measures

There are two types of cluster validity measures, namely set-based and pairwise-based measures[1]. These two types of measures can be extended in different manners.

3.1 Extension of Set-Based Cluster Validity Measures

First, the way to extend set-based cluster validity measures[10,12] such as cluster purity and entropy are described in this section. The properties of each measure were studied in the literature[1].

By considering neighborhood relation of clusters, the neighbor class distribution should be taken into account to the degree of certain class contained in a cluster, that is, the data points of the same class in the neighbor clusters should have a high weight, while those of distant clusters should have a low weight based on the inter-cluster distance as the diagram is shown in Fig. 1.

Let $f(\boldsymbol{u}; l)$ be a density distribution of class label $l \in \mathbf{T}$ at $\boldsymbol{u} \in \boldsymbol{\Omega}$, where $\boldsymbol{\Omega}$ denotes a data space, and $h(\boldsymbol{u}, \boldsymbol{v}) : \boldsymbol{\Omega} \times \boldsymbol{\Omega} \mapsto \mathbb{R}$ be a weighting function based on the neighborhood relation. Based on the above concept, a class density distribution $f(\boldsymbol{u}; l)$, a data density distribution $f(\boldsymbol{u})$, and a total volume of data N are smoothed by the weighting function $h(\boldsymbol{u}, \boldsymbol{v})$ as follows:

$$\hat{f}(\boldsymbol{u}; l) = \int_{\boldsymbol{\Omega}} h(\boldsymbol{u}, \boldsymbol{v}) f(\boldsymbol{v}; l) dv, \tag{1}$$

$$\hat{f}(\boldsymbol{u}) = \sum_{l \in \mathbf{T}} \hat{f}(\boldsymbol{u}; l) = \sum_{l \in \mathbf{T}} \int_{\boldsymbol{\Omega}} h(\boldsymbol{u}, \boldsymbol{v}) f(\boldsymbol{v}; l) dv, \tag{2}$$

[1] This work introduces the smoothing function into several cluster validity measures, in actual use, a measure should be selected according to the target application and the aspects the user wants to evaluate.

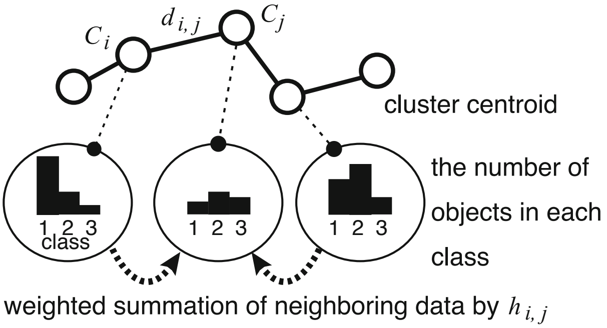

C_i
$d_{i,j}$
C_j
cluster centroid
the number of objects in each class
1 2 3
class
1 2 3
1 2 3
weighted summation of neighboring data by $h_{i,j}$

Fig. 1. Extension of a set-based clustering measure. The basic statistics are weighted by the neighborhood relation based on inter-cluster distance $d_{i,j}$. This example shows topology-based distance.

$$\hat{N} = \int_{\mathbf{\Omega}} \hat{f}(\boldsymbol{u})du = \int_{\mathbf{\Omega}} \sum_{l \in \mathbf{T}} \int_{\mathbf{\Omega}} h(\boldsymbol{u}, \boldsymbol{v}) f(\boldsymbol{v}; l) dv du. \tag{3}$$

Discretizing eqs. (1) to (3), let $N_{l,i}$ be the number of objects with class l in the i^{th} cluster $C_i \in \mathbf{C}$; $N_{l,i} = \#\{\boldsymbol{x}_k | t(k) = l, c(k) = C_i\}$, where # denotes the number of elements. N_i denotes the number of objects in cluster C_i; $N_i = \#\{\boldsymbol{x}_k | c(k) = C_i\}$. Also N denotes the total number of objects; $N = \#\{\boldsymbol{x}_k | \boldsymbol{x}_k \in \mathbf{S}\}$. Eqs. (1) to (3) can be rewritten as follows:

$$N'_{l,i} = \sum_{C_j \in \mathbf{C}} h_{i,j} N_{l,j}, \tag{4}$$

$$N'_i = \sum_{l \in \mathbf{T}} N'_{t,i} = \sum_{l \in \mathbf{T}} \sum_{C_j \in \mathbf{C}} h_{i,j} N_{l,j}, \tag{5}$$

$$N' = \sum_{C_i \in \mathbf{C}} N'_i = \sum_{C_i \in \mathbf{C}} \sum_{l \in \mathbf{T}} \sum_{C_j \in \mathbf{C}} h_{i,j} N_{l,j}. \tag{6}$$

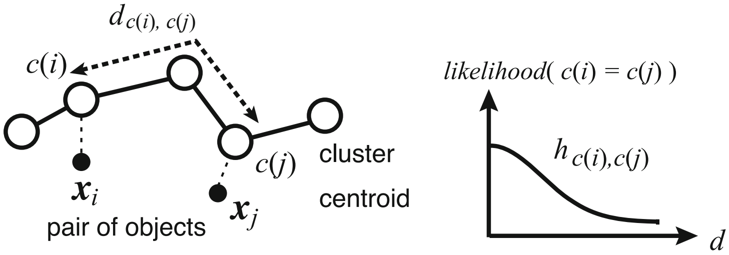
Here, $h_{i,j}$ can be used any monotonically decreasing function, for example, the often encountered Gaussian function: $h_{i,j} = \exp(-d_{i,j}/\sigma^2)$, where $d_{i,j}$ denotes inter-cluster distance and $\sigma(>0)$ is a smoothing (neighborhood) radius.

Thus, weighted cluster purity and entropy, for example, are defined using the weighted statistics of eqs. (4), (5), and (6) as follows:

weighted Cluster Purity (wCP)

$$\mathrm{wCP}(\mathbf{C}) = \frac{1}{N'} \sum_{C_i \in \mathbf{C}} \max_{l \in \mathbf{T}} N'_{l,i}. \tag{7}$$

The original purity is an average of the ratio that a majority class occupies in each cluster, whereas in the weighted purity a majority class is determined by the neighbor class distribution $\{N'_{l,i}\}$.



(a) Distance of data pair on a graph (b) Likelihood function

Fig. 2. Extension of a pairwise-based clustering measure. A likelihood function is introduced to represent a degree that a data pair belongs to the same cluster.

weighted Entropy (wEP)

$$\mathrm{wEP}(\mathbf{C}) = \frac{1}{|\mathbf{C}|} \sum_{C_i \in \mathbf{C}} \mathit{Entropy}(C_i), \tag{8}$$

$$\mathit{Entropy}(C_i) = -\frac{1}{\log N'} \sum_{l \in \mathbf{T}} \frac{N'_{l,i}}{N'_i} \log \frac{N'_{l,i}}{N'_i}, \tag{9}$$

where $|\mathbf{C}|$ denotes a cluster number. The original entropy indicates the degree of unevenness of class distribution within a cluster, whereas the extended entropy includes unevenness of the neighboring clusters.

3.2 Extension of Pairwise-Based Cluster Validity Indices

This section describes an extension of pairwise-based cluster validity measures[1,14]. Table 1 shows a class and cluster confusion matrix of data pairs, where a, b, c, d are the number of data pairs where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ do or do not belong to the same class/cluster.

Table 1. Class and cluster confusion matrix of data pairs

	$t(i) = t(j)$	$t(i) \neq t(j)$
$c(i) = c(j)$	a	b
$c(i) \neq c(j)$	c	d

Here, we introduce $likelihood(c(i) = c(j))$ indicating a degree that a data pair $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belongs to the same cluster instead of the actual number of data pairs. The likelihood is given by the inter-cluster distance of the data pair as shown in Fig. 2(a). The same weighting function as in sec. 3.1 is available for the likelihood function (Fig. 2(b)); $likelihood(c(i) = c(j)) = h_{c(i),c(j)}$. Then, a, b, c, d are replaced by summation of the likelihoods as follows:

$$a' = \sum_{\{i,j|t(i)=t(j)\}} h_{c(i),c(j)}, \tag{10}$$

$$b' = \sum_{\{i,j|t(i)\neq t(j)\}} h_{c(i),c(j)}, \tag{11}$$

$$c' = \sum_{\{i,j|t(i)=t(j)\}} \left(1 - h_{c(i),c(j)}\right) = a + c - a', \tag{12}$$

$$d' = \sum_{\{i,j|t(i)\neq t(j)\}} \left(1 - h_{c(i),c(j)}\right) = b + d - b'. \tag{13}$$

With these extended a', b', c' and d', weighted pairwise accuracy and pairwise F-measure are defined as follows:

weighted Pairwise Accuracy (wPA)

$$\text{wPA}(\mathbf{C}) = \frac{a' + d'}{a' + b' + c' + d'}. \tag{14}$$

The original pairwise accuracy is a ratio of the number of pairs in the same class belonging to the same cluster, or the number of pairs in different classes belonging to different clusters, against all pairs. The weighted PA is the degree to which pairs in the same class belong to the neighbor clusters or that pairs in different classes belong to distant clusters.

weighted Pairwise F-measure (wPF)

$$\text{wPF}(\mathbf{C}) = \frac{2 \cdot P \cdot R}{P + R}, \tag{15}$$

where $P = a'/(a'+b')$ is precision, that is a measure of the same class among each cluster, and $R = a'/(a'+c')$ is recall that is a measure of the same cluster among each class. The original pairwise F-measure is a harmonic average of the precision and the recall. While, the weighted PF is based on a degree that the data pairs belong to the same cluster.

3.3 Weighting Function

For the weighting function for smoothing in the set-based and the likelihood in pairwise-based measures, any monotonically decreasing function $h_{i,j} \geq 0$ is feasible, including the Gaussian or a rectangle function. Note that the extended measures are exactly the same as the original measures when $h_{i,j} = \delta_{i,j}$ (δ is the Kronecker delta).

The neighborhood radius effects the degree of smoothing and likelihood. Fig. 3 illustrates that the measure evaluates individual clusters, that is the original values, as the radius becomes zero ($\sigma \to 0$). On the other hand, as the radius becomes larger ($\sigma \to \infty$), the data space is smoothed by almost the same weights, and all micro-clusters are treated as one big cluster. The way to find the optimal radius is described in section 3.4.

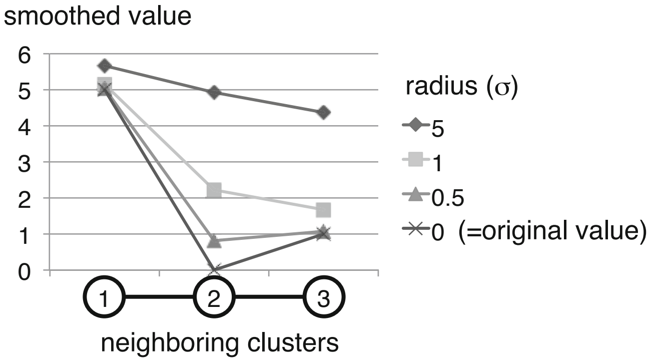


Fig. 3. Example of the effect of smoothing radius. Values over the neighborhood relation of the clusters become smoother as the radius increases.

3.4 Optimal Smoothing Radius

Our smoothed measures include a neighborhood relation in the conventional cluster validity measures. In order to evaluate a *neighbor component* within the measure, we defined as:

Definition 4. *(neighbor component) The quantity within the smoothed cluster validity value (Eval) that are caused by the neighborhood relation.*

Here, $Eval$ refers to the output value of wCP, wEP, wPA, or wPF in this paper.

Then, the neighbor component (NC) can be computed by comparing $Eval$s with randomized neighborhood relation.

$$NC(\sigma) = |Eval - \lim_{n\to\infty} Eval_{rnd(n)}| = |Eval(d_{i,j}) - Eval(\bar{d}_{i,j}))|, \tag{16}$$

where $Eval_{rnd(n)}$ denotes an average of $Eval$ when inter-cluster distances are n times shuffled, and when $n \to \infty$ this value converges to $Eval$ with the average of all inter-cluster distances $\bar{d}_{i,j}$. It is assumed that the smoothing radius that maximizes the neighbor component is the optimal one, i.e., $\sigma^* = \arg\max_\sigma NC(\sigma)$. Then, the optimal evaluation value can be $Eval^* = Eval(\sigma^*)$.

4 Evaluation of the Smoothed Validity Measures

This section describes the experiment to clarify the properties of the proposed smoothed validity measures.

4.1 Settings of Clustering and Neighborhood Relation

1. **kmc-knn**
   Typical k-means clustering was used to produce a clustering and mutual k-nearest neighbor (kmc-knn) was used to obtain the neighbor relation. With parameters of the prototype (micro-cluster) number $k1$ and of nearest neighbors $k2$, adjacent matrix $\mathbf{A} = (a_{i,j})$ can be given by:

$$a_{i,j} = \begin{cases} 1 & \text{if } C_j \in \mathbf{O}(C_i) \text{ and } C_i \in \mathbf{O}(C_j) \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

   where $\mathbf{O}(C_i)$ denotes a set of k-nearest neighbor clusters from C_i, where $d(C_i, C_j)$ is given by Euclidean distance. Then, distance matrix $\mathbf{D} = (d_{i,j})$ can be given by topological distance, in this experiment the shortest path between C_i and C_j is used, where the shortest path of all pairs are calculated by Warshall-Floyd Algorithm.
2. **SOM**
   Also the SOM[6] was used as an another type of producing micro-cluster prototypes with neighbor relation. In the SOM, the neurons of prototypes correspond to centroids of micro-clusters. The standard batch type SOM is used in this work. A distance matrix $\mathbf{D}$ is given by $d_{i,j} = ||\boldsymbol{r}_i - \boldsymbol{r}_j||$, where $\boldsymbol{r}$ is a coordinate of a neuron within the topology space of the SOM.

4.2 Datasets

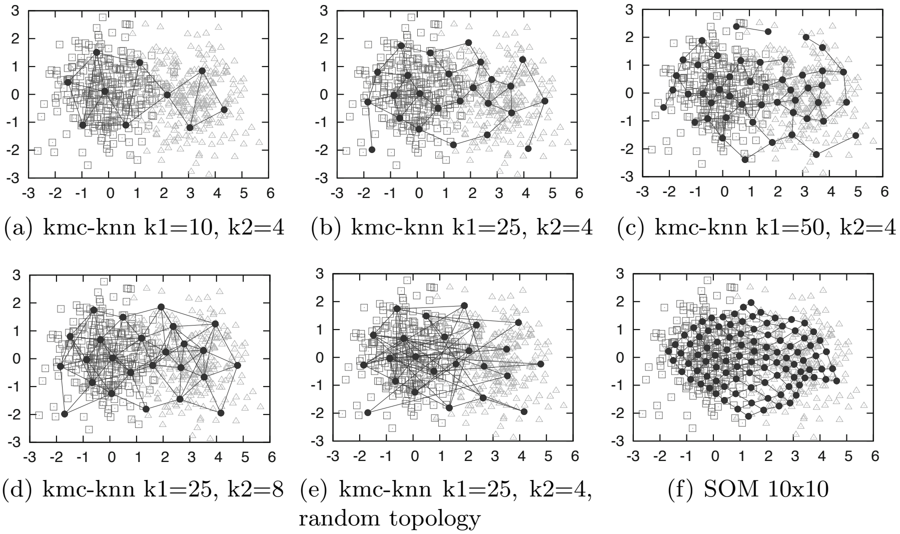
1. **Synthetic data**
   In order to evaluate the proposed measure, two classes of two-dimensional synthetic data were prepared, where 300 data points for each class were generated from different Gaussian distributions. The data distribution and examples of graphs are illustrated in Fig. 4.
2. **Real-world data**
   Well-known open datasets[2] were used as real-world data: Iris data (150 samples, 4 attributes, 3 classes), Wine data (178 samples, 13 attributes, 3 classes), and Glass Identification data (214 samples, 9 attributes, 6 classes).

4.3 Effect of Smoothing Radius - Finding the Optimal Radius

Fig. 5 shows the evaluation values of the smoothed validity measures for the synthetic data using kmc-knn. The larger value is the better except entropy. The values are average of 100 runs of randomized initial values.

Firstly, the total evaluation values ($Eval$) provides always better value than that of random topology ($Eval_{rnd}$) where neighborhood relation of the prototypes is destroyed. This means that the proposed measures evaluate both cluster validity and neighborhood relation of the clusters.

[2] http://archive.ics.uci.edu/ml/



(a) kmc-knn k1=10, k2=4 (b) kmc-knn k1=25, k2=4 (c) kmc-knn k1=50, k2=4

(d) kmc-knn k1=25, k2=8 (e) kmc-knn k1=25, k2=4, random topology (f) SOM 10x10

Fig. 4. Cluster prototypes (●) with topology-based neighbor relation on two dimensional synthetic data. The data points (□, △) were generated from two Gaussian distributions; $N(\mu_1, 1)$ and $N(\mu_2, 1)$, where $\mu_1 = (0, 0)$ and $\mu_2 = (3, 0)$.

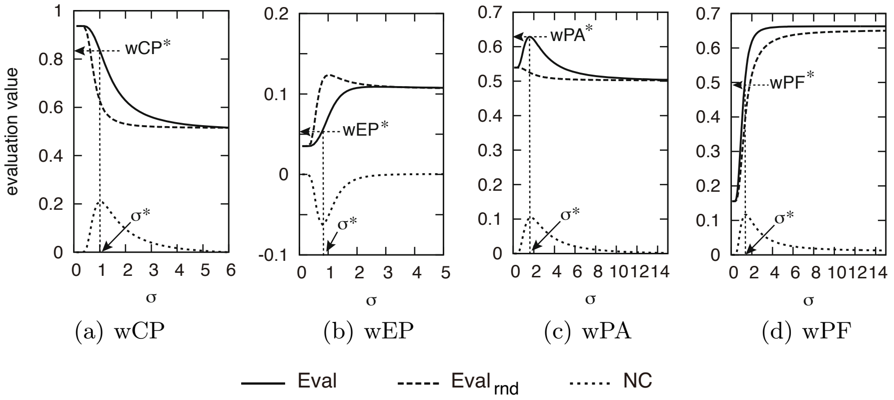
Secondly, as the smoothing radius becomes close to zero ($\sigma \to 0$), the extended measure evaluates individual clusters without neighborhood relation. Whereas, as the radius becomes larger ($\sigma \to \infty$), the extended measure treats whole data as one big cluster as mentioned before. Therefore, the solid and the broken lines gradually become equal as the radius becomes close to zero or becomes much larger.

Thirdly, the neighbor component has a monomodality against the radius in all measures, since there exists an appropriate radius to the average class distribution. Since the smoothed measure is a composition of cluster validity and neighborhood relation, the radius that gives the maximum *Eval* does not always match with that of neighbor component, for instance, wCP, wEP, and wPF in Fig. 5. Therefore, the neighbor component should be examined to find the appropriate radius. Also the appropriate radius depends on function of the measure such as purity, F-measure, or entropy. This means that the user should use different radius for each measure.

These three trends appear also in SOM (omitted due to page limitation).

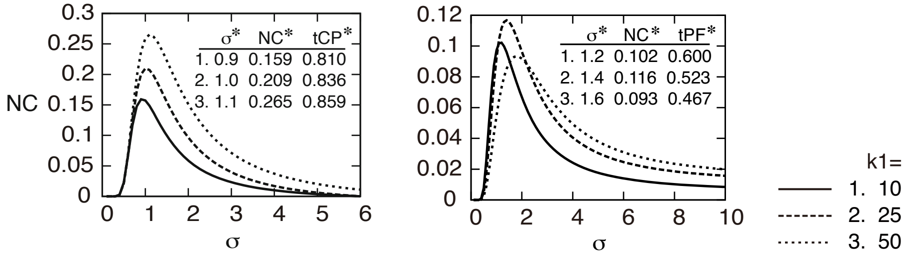
4.4 Effect of Prototype Number

The effect of prototype number is examined by changing $k1 = 10, 25, 50$ (Fig. 6). In wPF, $k1 = 25$ provides the highest neighbor component (0.116 at $\sigma = 1.4$) among three (Fig. 6(b)). wPF can suggest an optimal prototype number in terms of maximizing the neighbor component in the measure, which means neighbor



(a) wCP (b) wEP (c) wPA (d) wPF

Fig. 5. The effect of smoothing radius (synthetic data, kmc-knn($k1 = 25, k2 = 4$)); total evaluation value ($Eval$), $Eval$ with random topology ($Eval_{rnd}$), neighbor component (NC)



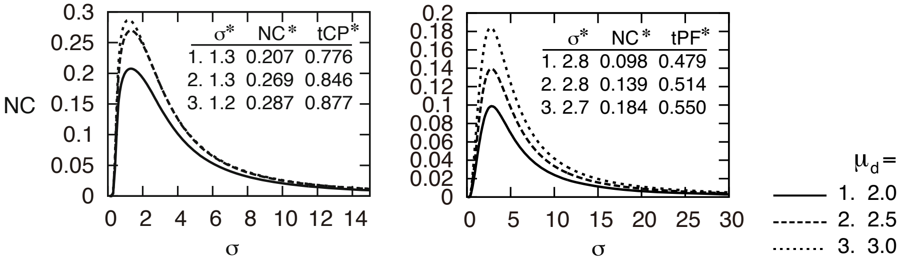
(a) wCP (b) wPF

Fig. 6. The effect of prototype number (synthetic data, kmc-knn($k2 = 4$)). The maximum neighbor component (NC^*) and total values (wCP and wPF) are listed together in the table.

relation of class distribution is maximized. However, the larger $k1$ the better in wCP (Fig. 6(a)). This is because the function of cluster purity given by eq. (7), that is, the smaller number of elements in some cluster tends to give better purity.

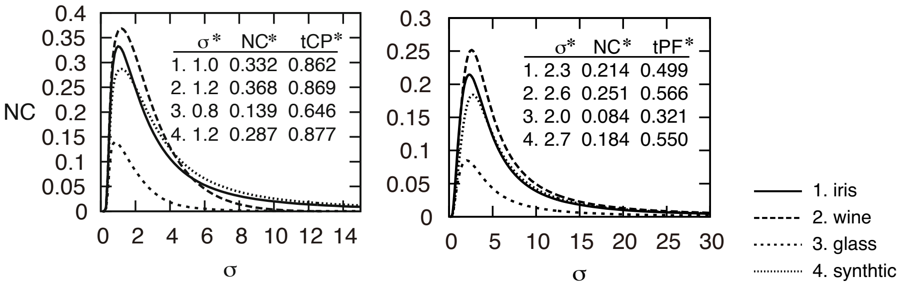
4.5 Effect of Class Overlap

The effect of class overlap is examined (Fig. 7) by changing distance between class centers $\mu_d = \mu_2^x - \mu_1^x$ from 2.0 to 3.0 in the synthetic data. Observing Fig. 7, the lower class overlap is, the better the neighbor component and the total values. However, the optimal radii are nearly the same even in different class overlap. This means that our measure can determine the optimal radius independent to class overlap, and can evaluate volume of overlap.



(a) wCP (b) wPF

Fig. 7. The effect of class overlap (synthetic data, SOM(10×10))



(a) wCP (b) wPF

Fig. 8. The effect of dataset, SOM (10×10)

4.6 Real-World Data

Fig. 8 shows the result for real-world data using SOM. Though there exists an optimal radius, the optimal radii vary depending on dataset, i.e., the number of classes and the class distribution. This result indicates that depending on dataset and measure, a user should use different radius that gives the maximum volume of neighbor component.

5 Conclusion

This paper proposed a novel and generic smoothed cluster validity measures based on neighborhood relation of clusters with external criteria. The experiments revealed the existence of an optimal neighborhood radius which maximizes the neighbor component. A user should use an optimal radius depending on a function of measure and a dataset. Our measure can determine the optimal radius independent to class overlap, and can evaluate volume of class overlap. In addition, feature selection, metric learning[13,15], and a correlation index for multilabels to determine the most relevant class are promising future directions for this work.

Acknowledgment. This work was supported by KAKENHI (21700165).

References

1. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval 699(12), 461–486 (2009)
2. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Transactions on Pattern Analsis and Machine Intelligence (TPAMI) 1(4), 224–227 (1979)
3. Deborah, L.J., Baskaran, R., Kannan, A.: A survey on internal validity measure for cluster validation. International Journal of Computer Science & Engineering Survey (IJCSES) 1(2), 85–102 (2010)
4. Dunn, J.C.: Well separated clusters and optimal fuzzy partitions. Journal of Cybernetics 4, 95–104 (1974)
5. Halkidi, M., Vazirgiannis, M.: Clustering validity assessment using multi representatives. In: Proc. 2nd Hellenic Conference on Artificial Intelligence, pp. 237–248 (2002)
6. Kohonen, T.: Self-Organizing Maps. Springer (1995)
7. Kovács, F., Legány, C., Babos, A.: Cluster validity measurement techniques. Engineering 2006, 388–393 (2006)
8. Kremer, H., Kranen, P., Jansen, T., Seidl, T., Bifet, A., Holmes, G., Pfahringer, B.: An effective evaluation measure for clustering on evolving data streams. In: Proc. the 17th SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2011), pp. 868–876 (2011)
9. Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J.: Understanding of internal clustering validation measures. In: Proc. IEEE International Conference on Data Mining (ICDM 2010), pp. 911–916 (2010)
10. Rendón, E., Abundez, I., Arizmendi, A., Quiroz, E.M.: Internal versus external cluster validation indexes. International Journal of Computers and Communications 5(1), 27–34 (2011)
11. Tasdemir, K., Merényi, E.: A new cluster validity index for prototype based clustering algorithms based on inter- and intra-cluster density. In: Proc. International Joint Conference on Neural Networks (IJCNN 2007), pp. 2205–2211 (2007)
12. Veenhuis, C., Koppen, M.: Data Swarm Clustering, ch. 10, pp. 221–241. Springer (2006)
13. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research (JMLR) 10, 207–244 (2009)
14. Xu, R., Wunsch, D.: Cluster Validity. Computational Intelligence, ch. 10, pp. 263–278. IEEE Press (2008)
15. Zha, Z.J., Mei, T., Wang, M., Wang, Z., Hua, X.S.: Robust distance metric learning with auxiliary knowledge. In: Proc. International Joint Conference on Artificial Intelligence (IJCAI 2009), pp. 1327–1332 (2009)

Sequential Entity Group Topic Model for Getting Topic Flows of Entity Groups within One Document

Young-Seob Jeong and Ho-Jin Choi

Department of Computer Science, KAIST
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Korea (South)
{pinode,hojinc}@kaist.ac.kr

Abstract. Topic mining is regarded as a powerful method to analyze documents, and topic models are used to annotate relationships or to get a topic flow. The research aim in this paper is to get topic flows of entities and entity groups within one document. We propose two topic models: Entity Group Topic Model (EGTM) and Sequential Entity Group Topic Model (S-EGTM). These models provide two contributions. First, topic distributions of entities and entity groups can be analyzed. Second, the topic flow of each entity or each entity group can be captured, through segments in one document. We develop collapsed gibbs sampling methods for performing approximate inference of the models. By experiments, we demonstrate the models by showing the analysis of topics, prediction performance, and the topic flows over segments in one document.

Keywords: Sequential topic model, Poisson-Dirichlet process, entity group.

1 Introduction

Analyzing documents on the Web is difficult due to the fast growing number of documents. Most of documents are not annotated, leading us to prefer unsupervised methods for analyzing document, and topic mining is one such method. This method is basically a probabilistic way to capture latent semantics, or topics, among documents. Since techniques like Probabilistic Latent Semantic Indexing (PLSI) [1] and Latent Dirichlet Allocation (LDA) [2] were first introduced, many studies have been derived from them: for example, to get relationships among entities in corpora [3, 4], to discover topic flows of documents in time dimension [5], or topic flows of segments in one document [6, 7], and so on. Capturing topic flows in one document (i.e., a fiction or a history) has special characteristics. For instance, adjacent segments in one document would influence each other because the full set of segments (i.e., the document) as a whole has some story. Moreover, the readers probably want to see the story in a perspective of each entity or each relationship. Although existing topic models tried to get topics of entity groups, no model has been proposed to obtain the topic flow of each entity or each relationship in one document. The topic flow in one document should also be useful for the readers to grasp the story easily.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 366–378, 2012.

In this paper, we propose two topic models, Entity Group Topic Model (EGTM) and Sequential Entity Group Topic Model (S-EGTM), claiming two contributions. First, topic distribution of each entity and of each entity group can be analyzed. Second, the topic flow of each entity and each relationship through segments in one document can be captured. To realize our proposal, we adopt collapsed gibbs sampling methods [8] to infer the parameters of the models.

The rest of the paper is organized as follows. In the following subsection, we preview the terminology to set out the basic concepts. Section 2 discusses related works. Section 3 describes out approach and algorithms in detail. Section 4 presents experiments and results. Finally, Section 5 concludes.

1.1 Terminology

In this subsection, we summarize the terminology used in this paper to clarify the basic concepts.

- **Entity:** Something which the user want to get information about it. It can be a name, an object, or even a concept such as love and pain.
- **Empty group (empty set):** A group having no entity.
- **Entity group:** A group having one or more entities.
- **Entity group size:** The number of entities in the entity group.
- **Entity pair:** A pair of two entities.
- **Topic (word topic):** A multinomial word distribution.
- **Entity topic:** A multinomial entity distribution of CorrLDA2.
- **Segment:** A part of a document. It can be a paragraph, or even a sentence.
- **Topic flow:** A sequence of topic distribution through segments of a document.
- **Relationship of entities:** A topic distribution of the entity group.

2 Related Work

In this section, we describe related studies with respect to *entity topic mining* and *sequential topic mining*.

The goal of *entity topic mining* is to capture the topic of each entity, or of each relationship of entities. Author Topic Model (ATM) [9] is a model for getting a topic distribution of each author. Although the model does not consider entities, it can be used for getting topics of entities by just considering an entity as an author. However, it does not involve a process of writing entities in the document. There are several studies about a model involving the process. The recent proposed model, named as Nubbi [4], tried to capture two kinds of topics, which are the word distributions of each entity and of each entity pair. However, since it takes two kinds of topics separately, the topics of entities will be different from that of entity pairs. Several topic models for analyzing entities were introduced in [3]. Especially, CorrLDA2 showed its best prediction performance. The model captures not only topics, but also entity topics. The entity topic is basically a list of entities, thus each entity topic plays a role as an entity group. This implies that it has a lack of capability of getting relationship of a certain entity group.

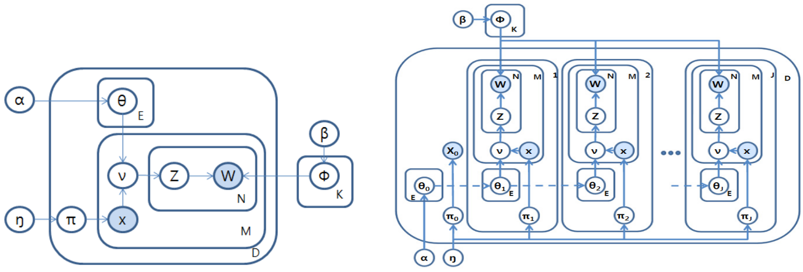
As for *sequential topic mining*, there are works which tried to get topic flows in different dimensions. Dynamic Topic Model (DTM) [5] aimed to capture topic flows of documents in time dimension. Probabilistic way to capture the topic patterns on weblogs, in both of space dimension and time dimension, was introduced in [10]. Multi-grain LDA (MG-LDA) [11] used topic distribution of each window in a document to get the ratable aspects. Although it utilizes sequent topic distributions to deal with multi-grained topics, the objective of the model is not getting a topic flow of the document. STM and Sequential LDA tried to get a topic flow within a document. The both studies are based on a nested extension of the two-parameter Poisson-Dirichlet Process (PDP). The STM assumes that each segment is influenced by the document, while the Sequential LDA assumes that each segment is influenced by its previous segment except for the first segment.

3 Sequential Entity Group Topic Model

Existing works on *entity topic mining* and *sequential topic mining*, however, cannot be used to obtain topic flow of each entity and each relationship within one document. The topic flow of each entity or each relationship should also be useful for the readers to grasp the story more easily. This section introduces two topic models, Entity Group Topic Model (EGTM) and Sequential Entity Group Topic Model (S-EGTM).

3.1 Entity Group Topic Model

A graphical model of EGTM is shown in Figure 1(a). The meaning of notations is described in Table 1. We suggest an assumption that a *relationship* of entities must influence the topic distribution of every corresponding *entity* and *entity group*. To apply the assumption into our model, we employ a power-set. For example, if an entity group, having two entities *A* and *B*, have a relationship, then the relationship influences the topics of its power set such as entity *A*, entity *B*, and *empty set*(*empty group*). Thus, a topic distribution of the empty set will be very similar to that of the document, because it associates with every sentence. Formally, the generative process is represented in Figure 2.



(a) (b)

Fig. 1. (a) Graphical model of EGTM. The colored circles represent the variables observable from the documents. (b) Graphical model of S-EGTM.

```
1. Draw a word distribution $\Phi$ from Dirichlet($\beta$)
2. For each document $d$,
   (1) For each entity group $e$,
       draw a topic distribution $\theta_{de}$ from Dirichlet($\alpha$
   (2) Draw an entity group dominance distribution $\pi_d$ from Dirichlet($\eta$)
   (3) For each sentence $s$,
       a. Choose an entity group $x_{ds}$ from Multinomial($\pi_d$)
       b. Given entity group $x_{ds}$, derive $v_{ds}$ by multiplying $\theta_{de}$ which are
          members of a power-set of the $x_{ds}$
       c. For each word $w$,
          (a) Choose a topic $z$ from Multinomial($v_{ds}$)
          (b) Given the topic $z$, generate a word $w$ from Multinomial($\Phi_z$)
```

Fig. 2. The formal generative process of EGTM

As a sentence has only one entity group or an entity, the size of power-set does not grow exponentially. If there is no observed entity in a sentence, then the sentence has an *empty group*. We developed a collapsed gibbs sampling. At each step of the Markov chain, the topic of the *i*th word is chosen using a conditional probability

$$P(z_i = k \mid \mathbf{z}',\mathbf{w},\mathbf{x}) \propto \frac{\beta_w + C_{kw}^{TW}}{\sum_v (\beta_v + C_{kv}^{TW})} \prod_{e \subset P(^{ds}\mathbf{x})} \frac{\alpha_k + C_{dek}^{DET}}{\sum_z (\alpha_z + C_{dez}^{DET})}. \tag{1}$$

The notations are described in Table 1, with a minor exceptional use of notation that and C_{kw}^{TW} in thiC_{dek}^{DET}expression exclude the *i*th word. Three parameters are obtained as follows:

$$\Phi_{kw} = \frac{\beta_w + C_{kw}^{TW}}{\sum_v (\beta_v + C_{kv}^{TW})}, \tag{2}$$

$$\theta_{dek} = \frac{\alpha_k + C_{dek}^{DET}}{\sum_z (\alpha_z + C_{dez}^{DET})}, \tag{3}$$

$$\pi_{de} = \frac{\eta_e + C_{de}^{DE}}{\sum_e (C_{de}^{DE} + \eta_e)}. \tag{4}$$

3.2 Sequential Entity Group Topic Model

A graphical model of S-EGTM is Figure 1(b). Formally, the generative process is represented in Figure 3. As S-EGTM gets a topic flow in a document, the *D* must be 1. A topic distribution of each segment is affected by that of previous segment, except that the first segment is affected by the document's topic distribution. To model this, we adopted Poisson-Dirichlet Process (PDP), as [7] does. If we use Chinese Restaurant Process (CRP) notations, then a word is a customer. The topics are dishes

Table 1. Meaning of the notations. The upper part contains variables for graphical models. The bottom part contains variables for representing the conditional probabilities.

Notations	Meaning of the notation
D	the number of documents
M	the number of sentences
N	the number of words
J	the number of segments
E	the number of unique entity groups
K	the number of topics
w	observed word
z	topic
ν	multiplying of multiple θ
x	observed entity group
θ	multinomial distribution over topics
Φ	multinomial distribution over words
π	multinomial distribution over entity groups
α	Dirichlet prior vector for θ
β	Dirichlet prior vector for Φ
ŋ	Dirichlet prior vector for π
a	a discount parameter for PDP
b	a strength parameter for PDP
z_i	the topic of *i*th word
′ (quote)	the situation that *i*th word is excepted
$\mathbf{z}$	the topic assignments for all words
e	an entity group
$\mathbf{w}$	a sequence of words of the document
$\mathbf{t}$	in document d, the sequence of vectors which have table counts for each topic
$\mathbf{T}_{dje}$	in segment *j* of document *d*, the number of tables associated with entity group *e*
$\mathbf{N}_{dje}$	in segment *j* of document *d*, the number of words associated with entity group *e*
t_{djez}	in segment *j* of document *d*, the number of tables of entity group *e*, which are assigned the topic *z*
n_{djez}	in segment *j* of document *d*, the number of words of entity group *e*, which are assigned the topic *z*
C_{kw}^{TW}	the number of words that are assigned the topic *k*
C_{dek}^{DET}	in the document *d*, the number of topics that appear in the sentence which the entity group *e* associates
C_{de}^{DE}	a frequency of the entity group *e* in the document *d*
$P(^{ds}\boldsymbol{x})$	the power-set of entity group of the sentence *s* in the document *d*
$S_{T,a}^{N}$	the generalized Stirling number. Intuitively, this is the number of cases that N customers seat on T tables in different sequence
$(b\,\vert\,a)_C$	the Pochhammer symbol with increment C
$(b)_C$	The Pochhammer symbol same as $(b\,\vert\,1)_C$
u_{dek}	An index of the first segment which has t_{duek}=0

and the segments are restaurants. The table count *t* is the number of tables occupied by customers. The customers sitting around a table share a dish. Especially, in nested PDP, the number of tables of next restaurant is a customer of current restaurant.

```
1. Draw a word distribution $\Phi$ from Dirichlet($\beta$)
2. For each document $d$,
   (1) For each entity group $e$, draw a topic distribution $\theta_{d0e}$ from Dirichlet($\alpha$)
   (2) Draw an entity group dominance distribution $\pi_{d0}$ from Dirichlet($\eta$)
   (3) Choose an entity group $x_{d0}$ from Multinomial($\pi_{d0}$)
   (4) For each segment $j$,
       a. Draw an entity group dominance distribution $\pi_{dj}$ from Dirichlet($\eta$)
       b. For each $e$, draw a topic distribution $\theta_{dje}$ from PDP($a$, $b$, $\theta_{d(j-1)e}$)
       c. For each sentence $s$,
          (a) Choose an entity group $x_{djs}$ from Multinomial($\pi_{dj}$)
          (b) Given entity group $x_{djs}$, derive $v_{djs}$ by multiplying $\theta_{dje}$
              which are members of a power-set of the $x_{djs}$
          (c) For each word $w$,
              i. Choose a topic $z$ from Multinomial($v_{djs}$)
              ii. Given the topic $z$, generate a word $w$ from Multinomial($\Phi_z$)
```

Fig. 3. The formal generative process of S-EGTM

When we do a collapsed gibbs sampling for topics, removing *i*th topic $z_{dgi}=k$ affects the table counts and topic distributions of entity group *e* in the segment *g*. Therefore, we need to consider three cases of conditional probabilities in terms of u_{dek}, as following.

First, when $u_{dek}=1$,

$$P(z_{dgi}=k\,|\,\mathbf{z}',\mathbf{w},\mathbf{x},\mathbf{t}) \propto \frac{\alpha_k + t'_{d1ek}}{\sum_z(\alpha_z + t'_{d1ez})}(b + aT'_{d1e})(\prod_{j=2}^{g}\frac{b + aT'_{dje}}{b + N_{d(j\;\;1)e} + T'_{dje}})\frac{\beta_w + C_{kw}^{TW}}{\sum_v(\beta_v + C_{kv}^{TW})}\,. \tag{5}$$

Second, when $1 < u_{dek} \leq g$,

$$P(z_{dgi}=k\,|\,\mathbf{z}',\mathbf{w},\mathbf{x},\mathbf{t}) \propto (\prod_{j=u_{dek}}^{g}\frac{b + aT'_{dje}}{b + N_{d(j\;\;1)e} + T'_{dje}})(\frac{S^{n_{d(u_{dk}\;\;1)ek}+1}_{t_{d(u_{dk}\;\;1)ek},a}}{S^{n_{d(u_{dk}\;\;1)ek}}_{t_{d(u_{dk}\;\;1)ek},a}})\frac{\beta_w + C_{kw}^{TW}}{\sum_v(\beta_v + C_{kv}^{TW})}\,. \tag{6}$$

Third, when $g < u_{dek}$,

$$P(z_{dgn}=k\,|\,\mathbf{z}',\mathbf{w},\mathbf{x},\mathbf{t}) \propto (\frac{S^{n'_{dgek}+1+t_{d(g+1)ek}}_{t'_{dgek},a}}{S^{n'_{dgek}+t_{d(g+1)ek}}_{t'_{dgek},a}})\frac{\beta_w + C_{kw}^{TW}}{\sum_v(\beta_v + C_{kv}^{TW})}\,. \tag{7}$$

The notations are described in Table 1, with a minor exceptional use of notation that in thiC_{kw}^{TW}expression exclude the *i*th word. At each step, we also sample a table count because a table count is affected by the number of words having the table’s topic and vice versa. If we assume that we remove a table count t_{dgek}, then new table count is sampled as follows:

$$P(t_{dgek} \,|\, \mathbf{z}', \mathbf{w}, \mathbf{x}, \mathbf{t}') \propto \left(\frac{\Gamma(\alpha_{\mathrm{k}} + t_{d1ek})}{\Gamma(\sum_{\mathrm{z}} \alpha_{\mathrm{z}} + t_{d1ez})}\right)^{\mathrm{g=1}} \left(\frac{S^{n_{d(g\ \ 1)ek} + t_{dgek}}_{t_{d(g\ \ 1)ek}, a}}{(b)_{N_{d(g\ \ 1)e} + T'_{dge}}}\right)^{\text{1-(g=1)}} \left((b \,|\, a)_{T'_{dge}} S^{n_{dgek} + t_{d(g+1)ek}}_{t_{dgek}, a}\right) \tag{8}$$

.

The notation g=1 means the first term is active only if it is the first segment, 1-(g=1) means the second term is active only if it is not the first segment. Γ is a gamma function. As considering every candidates of table count is intractable, we have to determine the window size of table count to consider. Among four parameters, we describe the approximate probabilities of two parameters as follows:

$$\theta_{d0ek} = \frac{\alpha_k + t_{d1ek}}{\sum_z (\alpha_z + t_{d1ez})}, \tag{9}$$

$$\theta_{djek} = \frac{n_{djek} - a \cdot t_{djek} + t_{d(j+1)ek} + \theta_{d(j-1)ek}(a \cdot T_{dje} + b)}{b + N_{dje} + T_{d(j+1)e}}. \tag{10}$$

4 Experiments

We used two data sets: the Bible and the fiction 'Alice'. We removed stop-words and did stemming by Porter stemmer. The sentences were recognized by '.', '?', '!', and "newline". After the deleting stop-words, the Bible has 295,884 words and the fiction 'Alice' has 11,605 words. As S-EGTM gets a topic flow in a document, it regards the Bible as a document consisting of 66 segments, and the fiction 'Alice' as a document consisting of 12 segments. In contrast, to compare EGTM with other models, we divided each document into separated files as segments. For every experiment, we set α=0.1, β=0.01, η=1, a=0.5, b=10, and the window size was 1.

4.1 The Size of Power-Set of Entity Groups

When we input a list of entities to consider, then a preprocessing will make a power-set hierarchy of existing entity groups in a document. Since a sentence is restricted to have an entity group, each entity group usually does not have more than three entities. Thus, as shown in Figure 4, the size of power-set does not grow exponentially. The used data is the Bible.

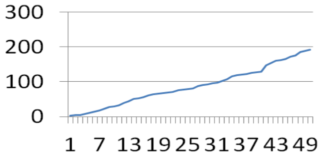


Fig. 4. The number of unique entity groups. The horizontal axis is the number of entities and the vertical axis represents the number of unique entity groups.

4.2 Topic Discovery

We used the Bible as a data and set the number of topics to be 20. We performed inference with 2,000 iterations. In Table 2 and Table 3, five topics of two models, LDA and EGTM, are shown. The obtained topics of the models are similar to each other because the empty group of EGTM associates with every sentence. The topics are coherent and specific to understand. EGTM additionally gives entity lists and relationship lists about each topic. With the lists, we can understand what are the topics that each entity or entity group associates with. For example, the topic *Mission work*, which is about missionary acts of apostles, is mostly handled in the *Act* written by *Paul* who lived in different era with *Abraham*. Nevertheless, the relationship {*God,Abraham*} has the topic *Mission work* the most, in the *Act*. This is caused by *Paul*'s writing about the covenant between *God* and *Abraham*. Since the covenant is that '*through your offspring all peoples on earth will be blessed*', the relationship {*God,Abraham*} has the topic *Mission work* in *Act*. Thus, EGTM helps us to grasp the documents in perspective of an entity or relationship.

Table 2. Topics obtained from LDA. The topic names are manually labeled. The listed chapters have a big proportion of the corresponding topic.

Topics	Gospel	Journey of Jesus & disciples	Mission work	Kingdom of Israel	Field life & Sanctuary
Top words	Christ faith love sin law spirit gospel grace church truth hope power dead	disciple father son crowd reply ask heaven truth answer kingdom Pharisee teacher law	Jew Jerusalem spirit holy sail Antioch prison apostle gentile Ship Asia travel province	king Israel Judah son temple reign Jerusalem father priest prophet Samaria altar servant	Egypt gold curtain Israelite cubit blue altar mountain ring acacia pole ephod tent
Chapters	Romans~ Jude	Matthew~ John	Acts	Kings	Exodus

4.3 Entity Prediction

We compared the EGTM with CorrLDA2 by entity prediction performance. We also made a model, named as *Entity-LDA*, which is a baseline. The Entity-LDA just counts the number of topics in sentences which have each entity, after LDA estimation. We used the Bible as data and varied the number of topics from 10 to 90. We used an entity list consisting of 16 entities: *God, Jesus, Petro, Judas, Paul, Mary, David, John, Abraham, Sarai, Solomon, Moses, Joshua, Aaron, Jeremiah,* and *Jonah*. For fair comparison, we made CorrLDA2 to use the entity list, rather than automatic Named Entity Recognition methods. For CorrLDA2, we set the number of entity topics same as the number of word topics, because we observed that the prediction results are similar with different numbers of entity topics. We did 10-fold cross validation for the comparison, and got the prediction results using the process in Figure 5.

Table 3. Topics obtained from EGTM. The topic names are manually labeled.

Topics	Gospel	Journey of Jesus & disciples	Mission work	Kingdom of Israel	Field life & Sanctuary
Top words	Christ	disciple	Jew	king	land
	faith	father	Jerusalem	Israel	Egypt
	love	son	holy	Judah	curtain
	law	crowd	spirit	temple	Israelite
	sin	reply	sail	Jerusalem	cubit
	grace	truth	ship	son	gold
	gospel	ask	gentile	reign	mountain
	world	Pharisee	speak	Samaria	altar
	spirit	kingdom	disciple	prophet	ring
	hope	teacher	believe	father	frame
	church	world	Christ	priest	blue
	life	heaven	Antioch	altar	tent
	boast	answer	prison	servant	pole
Chapters	Romans ~ Jude	Matthew ~ John	Acts	Kings	Exodus
Entities	God, Jesus, Paul, John	God, Jesus, Mary, Judas, David, Abraham, Joshua, Moses	God, Jesus, Paul, Judas, John, David, Abraham, Moses	God, David, Abraham, Solomon, Moses	God, Abraham, Moses, Joshua, Aaron
Relation-ships	{God,Jesus}, {God,Paul}, {God,John}, {Jesus,Paul}	{God,Jesus}, {Abraham,Jesus}, {Jesus,David}, {Abraham,Joshua, David}	{God,Jesus}, {Paul,Jesus}, {Paul,Judas}, {John,Jesus}, {David,Judas}, {Paul,John}, {God,Abraham}	{God,David}, {Solomon, David}, {God, Solomon}, {David,God, Solomon}	{God,Abraham}, {God,Moses, Abraham}, {Aaron,God}, {Moses,Joshua}, {Moses,Abraham }

1. Train the Entity-LDA, EGTM and CorrLDA2 with same training data.
2. For each sentence of test data, three models are supposed to choose one of 16 entities using the following predictive distributions:

   Entity-LDA : $P(e|s) \propto \prod_{w} \sum_{t} P(w|t)P(t|e)$,

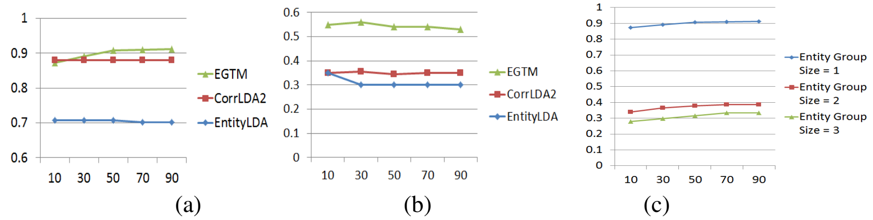
   CorrLDA2 : $P(e|s) \propto \prod_{w} \sum_{t} P(w|t)P(t|d)$ as in [3],

   EGTM : $P(e|s) \propto \prod_{w} \sum_{t} P(w|t)P(t|e)$

   where w is a word of a sentence s, d represents each training document, t is a topic, and e is the entity. Especially, $P(t|d)$ is obtained by resampling with $P(w|t)$.
3. The accuracy is the number of correct choices divided by the number of total choices.

Fig. 5. The process of the entity prediction

The test data consists of sentences which have at least one entity. If a sentence has multiple entities, then choosing one of them is regarded as a correct choice. As depicted in Figure 6(a), the CorrLDA2 shows fixed performance because the resampling makes *P(t|d)* to be fixed. EGTM outperforms other models because the topics of Entity-LDA have nothing to do with entities and CorrLDA2 does not get the topic distribution of each entity. The performance of EGTM grows as the number of topics grows because the Bible covers various topics. EGTM shows better performances than CorrLDA2 because of two reasons. First, CorrLDA2 does not directly get the topic distribution of each entity and it disperses the topic distribution of each entity into multiple entity topics. Second, CorrLDA2 takes data exclusively. To be specific, the data already used for entity topics will not be used for word topics.



(a) (b) (c)

Fig. 6. (a) The entity prediction performances of three models. The horizontal axis is the number of topics. The vertical axis means a prediction rate. (b) The entity pair prediction performances. (c) The entity group prediction performances.

4.4 Entity Pair Prediction

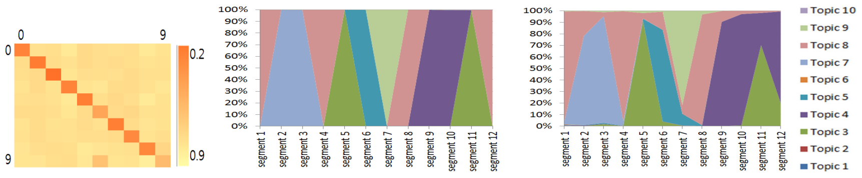
We compared the entity pair prediction performance between EGTM and CorrLDA2. For fair comparison, we used *entity-entity affinity* of [3]. The entity-entity affinity, defined as $P(e_i|e_j)/2+P(e_j|e_i)/2$, is to rank *true pairs* and *false pairs*. The true pairs exist in only unseen document, while the false pairs do not exist. The prediction performance is the number of true pairs in half of high ranked pairs, divided by the number of total pairs. We prepared 50 true pairs and 50 false pairs. The models have different methods to get $P(e_i|e_j)$ which is obtainable from $\sum_t P(e_i|t)P(t|e_j)$. Entity-LDA just counts the number of each topic. CorrLDA2 uses entity topic distributions. For examp$P(t/e_j)=\sum_{et}P(t|et)P(et|e_j)$where *et* means each entity topic. Figure 6(b) describes the prediction performance. Because the most entities of the Bible old testament usually do not appear in the Bible new testament, the overall prediction performances is low. EGTM outperforms CorrLDA2 and Entity-LDA, because EGTM directly takes a topic distribution of each entity.

4.5 Entity Group Prediction

We do not compare the prediction performance with other models because the other models lack ability to get topic distributions of entity groups. Instead, we demonstrate prediction performance with different entity group sizes. The predictive distribution is $P(eg/s) \propto \sum_d P(eg_d) \prod_w \sum_t P(w|t)P(t|eg_d)$, where *eg* represents the entity group, and *d* represents each training document. Figure 6(c) shows the prediction performance. The accuracy is the number of correct predictions divided by the number of total predictions. The prediction performance of smaller entity group is better than that of larger entity group, because it is harder to predict more entities.

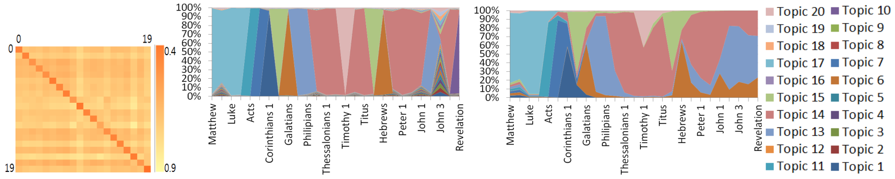
4.6 Topic Flow

We compare the topic flow of S-EGTM with the topic distributions of EGTM. To show the topic consistency between the two models, we trained S-EGTM boosted from the trained EGTM with 2,000 iterations. The Bible new testament and the fiction 'Alice' are used as data. We analyze the entity *Alice* with 10 topics, and analyze a relationship {*Jesus*, *God*} with 20 topics. Figure 7 and Figure 8 show the topic flows of the entity *Alice* and the relationship {*Jesus*, *God*}, respectively.



(a) (b) (c)

Fig. 7. (a) The confusion matrix by Hellinger distance, with the fiction 'Alice' as a data, where S-EGTM topics run along the Y-axis. (b) Topic flow of entity *Alice* by EGTM. (c) Topic flow of entity *Alice* by S-EGTM.



(a) (b) (c)

Fig. 8. (a) The confusion matrix by Hellinger distance, with the Bible new testament as a data, where S-EGTM topics run along the Y-axis. (b) Topic flow of relationship {*Jesus*, *Paul*} by EGTM. (c) Topic flow of relationship {*Jesus*, *Paul*} by S-EGTM.

Figure 7(a) and Figure 8(a) show the confusion matrices of the topic distributions generated by EGTM and S-EGTM. The diagonal cells are darker than others, meaning that the corresponding topics have low Hellinger distance. Thus, the topics of two models are consistent. Other than the Figure 7(a) and Figure 8(a), the horizontal axis means each segment, while the vertical axis represents topic proportion. Clearly, in Figure 7(b), each topic appears in totally different segments, which gives no idea about a topic flow through the segments. In contrast, in Figure

7(c), we can see the pattern that the topic 8(pink color) flows through every segment. As the topic 8 is about *Alice's tracking the rabbit*, its flow through every segment is coherent with the story. Consider the case of the relationship {*Jesus*, *God*} in more detail. In Figure 8(b), the topic *Gospel* (topic 14) is dominant in four separated parts, meaning that the relationship {*Jesus*, *God*} associates with the topic *Gospel* in only those separated four parts. This is caused by that the relationship has sparse topic distribution because it reflects only the sentences having the relationship. The separated appearance of the topic is not coherent with the Bible, because a purpose of the Bible new testament associates with the topic *Gospel* which is strongly about the news of the relationship {*Jesus*, *God*}. In contrast, in Figure 8(c), the topic *Gospel* appears like a flow from *Acts* to *Revelation*. This means the relationship {*Jesus*, *God*} associates with the topic *Gospel* without any cutting, through the segments. This is more coherent with the Bible. Thus, S-EGTM helps us to grasp the topic flow of an entity or a relationship by smoothing the sparse topic distribution of EGTM.

5 Conclusion

In this paper, we proposed two new generative models, Entity Group Topic Model (EGTM) and the Sequential Entity Group Topic Model (S-EGTM). S-EGTM reflects the sequential structure of a document in the hierarchical modeling. We developed collapsed gibbs sampling algorithms for the models. EGTM employs a power-set structure to get topics of entities or entity groups. S-EGTM is a sequential version of the EGTM, and employs nested two-parameter Poisson-Dirichlet process (PDP) to capture a topic flow over the sequence of segments in one document. We have analyzed the topics obtained from EGTM, and showed that topic flows generated by S-EGTM are coherent with the original document. Moreover, the experimental results show that the prediction performance of EGTM is better than that of CorrLDA2. Thus, we believed that the intended mechanisms of the EGTM and S-EGTM models work.

Acknowledgments. This work was supported by the National Research Foundation (NRF) grant (No. 2011-0018264) of Ministry of Education, Science and Technology (MEST) of Korea.

References

1. Hofmann, T.: Probabilistic Latent Semantic Indexing. In: SIGIR, pp. 50–57 (1999)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. In: NIPS, pp. 601–608 (2001)
3. Newman, D., Chemudugunta, C., Smyth, P.: Statistical entity-topic models. In: KDD, pp. 680–686 (2006)
4. Chang, J., Boyd-Graber, J.L., Blei, D.M.: Connections between the lines: augmenting social networks with text. In: KDD, pp. 169–178 (2009)
5. Blei, D.M., Lafferty, J.D.: Dynamic topic models. In: ICML, pp. 113–120 (2006)

6. Du, L., Buntine, W.L., Jin, H.: A segmented topic model based on the two-parameter Poisson-Dirichlet process. Machine Learning, 5–19 (2010)
7. Du, L., Buntine, W.L., Jin, H.: Sequential Latent Dirichlet Allocation: Discover Underlying Topic Structures within a Document. In: ICDM, pp. 148–157 (2010)
8. Griffiths, T.L., Steyvers, M.: Finding Scientific Topics. National Academy of Sciences, 5228–5235 (2004)
9. Rosen-Zvi, M., Griffiths, T.L., Steyvers, M., Smyth, P.: The Author-Topic Model for Authors and Documents. In: UAI, pp. 487–494 (2004)
10. Mei, Q., Liu, C., Su, H., Zhai, C.: A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In: WWW, pp. 533–542 (2006)
11. Titov, I., McDonald, R.T.: Modeling Online Reviews with Multi-grain Topic Models. CoRR (2008)

Topological Comparisons of Proximity Measures

Djamel Abdelkader Zighed, Rafik Abdesselam, and Asmelash Hadgu

Department of Computer Science and Statistics, ERIC laboratory,
University Lumiére of Lyon 2, Campus Porte des Alpes, France
{abdelkader.zighed,rafik.abdesselam}@univ-lyon2.fr,
asmelashtk@gmail.com

Abstract. In many fields of application, the choice of proximity measure directly affects the results of data mining methods, whatever the task might be: clustering, comparing or structuring of a set of objects. Generally, in such fields of application, the user is obliged to choose one proximity measure from many possible alternatives. According to the notion of equivalence, such as the one based on preordering, certain proximity measures are more or less equivalent, which means that they should produce almost the same results. This information on equivalence might be helpful for choosing one such measure. However, the complexity $O(n^4)$ of this approach makes it intractable when the size n of the sample exceeds a few hundred. To cope with this limitation, we propose a new approach with less complexity $O(n^2)$. This is based on topological equivalence and it exploits the concept of local neighbors. It defines equivalence between two proximity measures as having the same neighborhood structure on the objects. We illustrate our approach by considering 13 proximity measures used on datasets with continuous attributes.

Keywords: proximity measure, pre-ordering, topological equivalence.

1 Introduction

In order to understand and act on situations that are represented by a set of objects, very often we are required to compare them. Humans perform this comparison subconsciously using the brain. In the context of artificial intelligence, however, we should be able to describe how the machine might perform this comparison. In this context, one of the basic elements that must be specified is the proximity measure between objects.

Certainly, application context, prior knowledge, data type and many other factors can help in identifying of the appropriate measure. For instance, if the objects to be compared are described by boolean vectors, we can restrict our comparisons to a class of measures specifically devoted to this data type. However, the number of candidate measures might still remain quite large. Can we consider that all those remaining are equivalent and just pick one of them at random? Or are there some that are equivalent and, if so, to what extent? This information might interest a user when seeking a specific measure. For instance, in information retrieval, choosing a given proximity measure is an important issue. We effectively know that the result of a query depends on the measure used. For this reason, users may wonder which one more useful? Very often, users

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 379–391, 2012.

try many of them, randomly or sequentially, seeking a ”suitable” measure. If we could provide a framework that allows the user to compare proximity measures and therefore identify those that are similar, they would no longer need to try out all measures.

The present study proposes a new framework for comparing proximity measures. We deliberately ignore the issue of the appropriateness of the proximity measure as it is still an open and challenging question currently being studied. Comparing proximity measures can be analyzed from different angles:

- Axiomatically, as in the works of [1], [2] and [7], where two measures are considered equivalent if they possess the same mathematical properties.
- Analytically, as in the works of [2], [3] and [7], where two measures are considered equivalent if one can be expressed as a function of the other.
- Emperically, as in [20], where two proximity measures are considered similar if, for a given set of objects, the proximity matrices brought about over the objects are somewhat similar. This can be achieved by means of statistical tests such as the Mantel test [13]. We can also deal with this issue using an approach based on preordonance [7][8][18], in which the common idea is based on a principle which says that two proximity measures are closer if the preorder induced in pairs of objects does not change. We will provide details of this approach later on.

Nevertheless, these approaches can be unified depending on the extent to which they allow the categorization of proximity measures. Thus, the user can identify measures that are equivalent from those that are less so [3][8].

In this paper, we present a new approach for assessing the similarity between proximity measures. Our approach is based on proximity matrices and hence belongs to empirical methods. We introduce this approach by using a neighborhood structure of objects. This neighborhood structure is what we refer to as the topology induced by the proximity measures. For two proximity measures u_i and u_j, if the topological graphs produced by both of them are identical, then this means that they have the same neighborhood graph and consequently, the proximity measures u_i and u_j are in topological equivalence. In this paper, we will refer to the degree of equivalence between proximity measures. In this way, we can calculate a value of topological equivalence between pairs of proximity measures which would be equal to 1 for perfect equivalence and 0 for total mismatch. According to these values of similarity, we can visualize how close the proximity measures are to each other. This visualization can be achieved by any clustering algorithm. We will introduce this new approach more formally and show the principal links identified between our approach and that based on preordonnance. So far, we have not found any publication that deals with the problem in the same way as we do here.

The present paper is organized as follows. In Section 2, we describe more precisely the theoretical framework and we recall the basic definitions for the approach based on induced preordonnance. In Section 3, we introduce our approach, topological equivalence. In section 4, we provide some results of the comparison between the two approaches, and highlight possible links between them. Further work and new lines of inquiry provided by our approach are detailed in Section 5, the conclusion. We also make some remarks on how this work could be extended to all kinds of proximity measures, regardless of the representation space: binary [2][7][8][26], fuzzy [3][28] or symbolic, [11][12].

2 Proximity Measures and Preordonnance

2.1 Proximity Measures

In this article we limit our work to proximity measures built on Rp. Nevertheless, the approach could easily be extended to all kinds of data: quantitative or qualitative. Let us consider a sample of n individuals $x, y, \ldots$ in a space of p dimensions. Individuals are described by continuous variables: $x = (x_1, \ldots, x_p)$. A proximity measure u between two individual points x and y is defined as follows:

$$\begin{aligned} u : R^p \times R^p &\longmapsto R \\ (x,y) &\longmapsto u(x,y) \end{aligned}$$

with the following properties, $\forall (x,y) \in R^p \times R^p$:
P1: $u(x,y) = u(y,x)$.
P2: $u(x,x) \leq u(x,y)$, P2': $u(x,x) \geq u(x,y)$.
P3: $\exists \alpha \in R$: $u(x,x) = \alpha$.

We can also define δ: $\delta(x,y) = u(x,y) - \alpha$ a proximity measure that satisfies the following properties, $\forall (x,y) \in R^p \times R^p$:

T1: $\delta(x,y) \geq 0$.
T2: $\delta(x,x) = 0$.
T3: $\delta(x,x) \leq \delta(x,y)$.
T4: $\delta(x,y) = 0 \Rightarrow \forall z \;\; \delta(x,z) = \delta(y,z)$.
T5: $\delta(x,y) = 0 \Rightarrow x = y$.
T6: $\delta(x,y) \leq \delta(x,z) + \delta(z,y)$.
T7: $\delta(x,y) \leq \max(\delta(x,z), \delta(z,y))$.
T8: $\delta(x,y) + \delta(z,t) \leq \max(\delta(x,z) + \delta(y,t), \delta(x,t) + \delta(y,z))$.

A proximity measure that verifies properties T1, T2 and T3 is a dissimilarity measure. If it satisfies the properties T5 and T6 it becomes a distance. As shown in [1], there are some implications between these properties: T7 $\Rightarrow T6 \Leftarrow T8$

In Table 1, we give a list of 13 conventional proximity measures.

For our experiments and comparisons, we took many datasets from the UCI-repository and we carried out a lot of sub sampling on individuals and variables. Table 4 shows the datasets used in this work.

2.2 Preorder Equivalence

Two proximity measures, u_i and u_j generally lead to different proximity matrices. Can we say that these two proximity measures are different just because the resulting matrices have different numerical values? To answer this question, many authors,[7][8][18], have proposed approaches based on preordonnance defined as follows:

Table 1. Some proximity measures

MEASURE	SHORT	FORMULA
EUCLIDEAN	EUC	$u_E(x,y) = \sqrt{\sum_{j=1}^{p}(x_j - y_j)^2}$
MAHALANOBIS	MAH	$u_{Mah}(x,y) = \sqrt{(x-y)^t \Sigma^{-1}(x-y)}$
MANHATTAN	MAN	$u_{Man}(x,y) = \sum_{j=1}^{p} \lvert x_j - y_j \rvert$
MINKOWSKI	MIN	$u_{Min_\gamma}(x,y) = (\sum_{j=1}^{p} \lvert x_j - y_j \rvert^\gamma)^{\frac{1}{\gamma}}$
TCHEBYTCHEV	TCH	$u_{Tch}(x,y) = \max_{1 \leq j \leq p} \lvert x_j - y_j \rvert$
COSINE DISSIMILARITY	COS	$u_{Cos}(x,y) = 1 - \frac{<x,y>}{\lVert x \rVert \lVert y \rVert}$
CANBERRA	CAN	$u_{Can}(x,y) = \sum_{j=1}^{p} \frac{\lvert x_j - y_j \rvert}{\lvert x_j \rvert + \lvert y_j \rvert}$
SQUARED CHORD	SC	$u_{SC}(x,y) = \sum_{j=1}^{p}(\sqrt{x_j} - \sqrt{y_j})^2$
WEIGHTED EUCLIDEAN	WE	$u_{WE}(x,y) = \sqrt{\sum_{j=1}^{p} \alpha_i (x_j - y_j)^2}$
CHI-SQUARE	χ^2	$u_{\chi^2}(x,y) = \sum_{j=1}^{p} \frac{(x_j - m_j)^2}{m_j}$
JEFFREY DIVERGENCE	JD	$u_{JD}(x,y) = \sum_{j=1}^{p}(x_j \log \frac{x_j}{m_j} + y_j \log \frac{y_j}{m_j})$
PEARSON'S CORRELATION	ρ	$u_\rho(x,y) = 1 - \lvert \rho(x,y) \rvert$
NORMALIZED EUCLIDEAN	NE	$u_{NE}(x,y) = \sqrt{\sum_{j=1}^{p}(\frac{x_j - y_j}{\sigma_j})^2}$

Where p is the dimension of space, $x = (x_j)_{j=1,\ldots,p}$ and $y = (y_j)_{j=1,\ldots,p}$ two points in R^p, $(\alpha_j)_{j=1,\ldots,p} \geq 0$, Σ^{-1} the inverse of the variance and covariance matrix, σ_j^2 the variance, $\gamma > 0$, $m_j = \frac{x_j + y_j}{2}$ and $\rho(x,y)$ denotes the linear correlation coefficient of Bravais-Pearson.

Definition 1. *Equivalence in preordonnance: Let us consider two proximity measures u_i and u_j to be compared. If for any quadruple (x,y,z,t), we have: $u_i(x,y) \leq u_i(z,t) \Rightarrow u_j(x,y) \leq u_j(z,t)$, then, the two measures are considered equivalent.*

This definition has since reproduced in many papers such as [2], [3], [8] and [28]. This definition leads to an interesting theorem which is demonstrated in [2].

Theorem 1. *Equivalence in preordonnance: with two proximity measures u_i and u_j, if there is a strictly monotonic function f such that for every pair of objects (x,y) we have: $u_i(x,y) = f(u_j(x,y))$, then u_i and u_j induce identical preorder and therefore they are equivalent. The converse is also true.*

In order to compare proximity measures u_i and u_j, we need to define an index that could be used as a similarity value between them. We denote this by $S(u_i,u_j)$. For example, we can use the following similarity index which is based on preordonnance.

$$S(u_i,u_j) = \frac{1}{n^4} \sum_x \sum_y \sum_z \sum_t \delta_{ij}(x,y,z,t)$$

$$\text{where} \quad \delta_{ij}(x,y,z,t) = \begin{cases} 1 \text{ if } [u_i(x,y) - u_i(z,t)] \times [u_j(x,y) - u_j(z,t)] > 0 \\ \quad \text{or } u_i(x,y) = u_i(z,t) \text{ and } u_j(x,y) = u_j(z,t) \\ 0 \text{ otherwise} \end{cases}$$

S varies in the range [0, 1]. Hence, for two proximity measures u_i and u_j, a value of 1 means that the preorder induced by the two proximity measures is the same and therefore the two proximity matrices of u_i and u_j are equivalent.

The workflow in Fig 1 summarizes the process that leads to the similarity matrix between proximity measures.

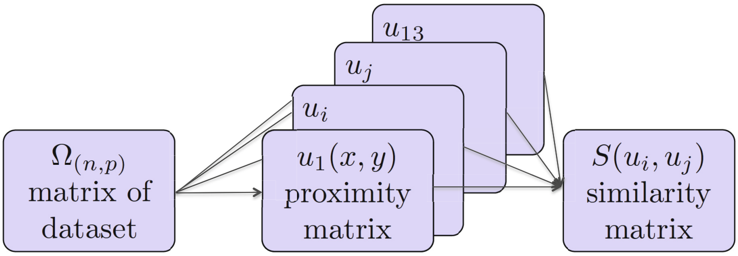

$\Omega_{(n,p)}$ matrix of dataset
$u_1(x,y)$ proximity matrix
u_i
u_j
u_{13}
$S(u_i,u_j)$ similarity matrix

Fig. 1. Workflow of preorder equivalence

As an example, in Table 2 we show the similarity matrix between the 13 proximity measures. This is the result of the work flow on the iris dataset.

Table 2. Preordonnance similarities: $S(u_i,u_j)$

S	u_E	u_{Mah}	u_{Man}	u_{Min_γ}	u_{Tch}	u_{Cos}	u_{Can}	u_{SC}	u_{WE}	u_{χ^2}	u_{JD}	u_ρ	u_{NE}
u_E	1												
u_{Mah}	.713	1											
u_{Man}	.966	.709	1										
u_{Min_γ}	.987	.712	.955	1									
u_{Tch}	.954	.694	.927	.965	1								
u_{Cos}	.860	.698	.848	.864	.857	1							
u_{Can}	.889	.678	.888	.886	.869	.861	1						
u_{SC}	.947	.703	.935	.946	.926	.880	.932	1					
u_{WE}	1	.713	.966	.987	.954	.860	.889	.947	1				
u_{χ^2}	.951	.705	.939	.950	.930	.881	.930	.995	.951	1			
u_{JD}	.949	.704	.937	.947	.928	.880	.931	.998	.949	.997	1		
u_ρ	.857	.682	.845	.862	.856	.940	.839	.865	.857	.866	.865	1	
u_{NE}	.911	.751	.915	.905	.882	.838	.872	.898	.911	.901	.899	.830	1

The comparison between indices of proximity measures has also been studied by [19], [20] from a statistical perspective. The authors proposed an approach that compares similarity matrices, obtained by each proximity measure, using Mantel's test [13], in a pairwise manner.

3 Topological Equivalence

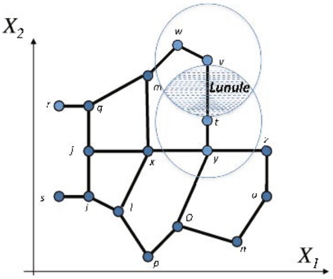
This approach is based on the concept of a topological graph which uses a neighborhood graph. The basic idea is quite simple: we can associate a neighborhood graph to each proximity measure (this is -our topological graph-) from which we can say that two proximity measures are equivalent if the topological graphs induced are the same. To evaluate the similarity between proximity measures, we compare neighborhood graphs and quantify to what extent they are equivalent.

3.1 Topological Graphs

For a proximity measure u, we can build a neighborhood graph on a set of individuals where the vertices are the individuals and the edges are defined by a neighborhood relationship property. We thus simplify have to define the neighborhood binary relationship between all couples of individuals. We have plenty of possibilities for defining this relationship. For instance, we can use the definition of the Relative Neighborhood Graph [16], where two individuals are related if they satisfy the following property:

If $u(x,y) \leq \max(u(x,z),u(y,z)); \forall z \neq x, \neq y$ then, $V_u(x,y) = 1$ otherwise $V_u(x,y) = 0$.

Geometrically, this property means that the hyper-lunula (the intersection of the two hyper-spheres centered on two points) is empty. The set of couples that satisfy this property result in a related graph such as that shown in Figure 2. For the example shown, the proximity measure used is the Euclidean distance. The topological graph is fully defined by the adjacency matrix as in Figure 2.



$$\left(\begin{array}{c|c}
V_u & \ldots\ \mathrm{x}\ \mathrm{y}\ \mathrm{z}\ \mathrm{t}\ \mathrm{u}\ \ldots \\
\hline
\vdots & \vdots\ \ \vdots\ \vdots\ \vdots\ \vdots\ \vdots\ \ldots \\
\mathrm{x} & \ldots\ 1\ 1\ 0\ 0\ 0\ \ldots \\
\mathrm{y} & \ldots\ 1\ 1\ 1\ 1\ 0\ \ldots \\
\mathrm{z} & \ldots\ 0\ 1\ 1\ 0\ 1\ \ldots \\
\mathrm{t} & \ldots\ 0\ 1\ 0\ 1\ 0\ \ldots \\
\mathrm{u} & \ldots\ 0\ 0\ 1\ 0\ 1\ \ldots \\
\vdots & \vdots\ \ \vdots\ \vdots\ \vdots\ \vdots\ \vdots\ \ldots
\end{array}\right)$$

Fig. 2. Topological graph built on RNG property

In order to use the topological approach, the property of the relationship must lead to a related graph. Of the various possibilities for defining the binary relationship, we can use the properties in a Gabriel Graph or any other algorithm that leads to a related graph such as the Minimal Spanning Tree, MST. For our work, we use only the Relative Neighborhood Graph, RNG, because of the relationship there is between those graphs [16].

3.2 Similarity between Proximity Measures in Topological Frameworks

From the previous material, using topological graphs (represented by an adjacency matrix), we can evaluate the similarity between two proximity measures via the similarity

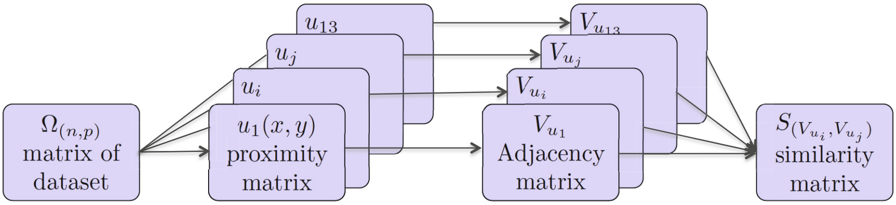


Fig. 3. Workflow of topological equivalence

between the topological graphs each one produces. To do so, we just need the adjacency matrix associated with each graph. The workflow is represented in Figure 3.

Note that V_{u_i} and V_{u_j} are the two adjacency matrices associated with both proximity measures. To measure the degree of similarity between the two proximity measures, we just count the number of discordances between the two adjacency matrices. The value is computed as:

$$S(V_{u_i},V_{u_j}) = \frac{1}{n^2}\sum_{x\in\Omega}\sum_{y\in\Omega}\delta_{ij}(x,y) \quad \text{where} \quad \delta_{ij}(x,y) = \begin{cases} 1 \text{ if } V_{u_i}(x,y) = V_{u_j}(x,y) \\ 0 \text{ otherwise} \end{cases}$$

S is the measure of similarity which varies in the range $[0,1]$. A value of 1 means that the two adjacency matrices are identical and therefore the topological structure induced by the two proximity measures in the same, meaning that the proximity measures considered are equivalent. A value of 0 means that there is a full discordance between the two matrices ($V_{u_i}(x,y) \neq V_{u_j}(x,y)\ \forall\omega \in \Omega^2$). S is thus the extent of agreement between the adjacency matrices. The similarity values between the 13 proximity measures in the topological framework for iris are given in Table 3.

Table 3. Topology similarities: $S(u_i,u_j)$

S	u_E	u_{Mah}	u_{Man}	u_{Min_γ}	u_{Tch}	u_{Cos}	u_{Can}	u_{SC}	u_{WE}	u_{χ^2}	u_{JD}	u_ρ	u_{NE}
u_E	1												
u_{Mah}	.978	1											
u_{Man}	.988	.974	1										
u_{Min_γ}	.998	.977	.987	1									
u_{Tch}	.980	.966	.971	.982	1								
u_{Cos}	.973	.972	.968	.973	.959	1							
u_{Can}	.982	.975	.984	.981	.967	.971	1						
u_{SC}	.989	.979	.984	.987	.973	.974	.988	1					
u_{WE}	1	.978	.988	.998	.980	.973	.982	.989	1				
u_{χ^2}	.989	.979	.984	.987	.973	.974	.988	1	.989	1			
u_{JD}	.989	.979	.984	.987	.973	.974	.988	1	.989	1	1		
u_ρ	.971	.971	.967	.970	.958	.980	.969	.971	.971	.971	.971	1	
u_{NE}	.985	.979	.984	.984	.971	.972	.983	.985	.985	.985	.985	.970	1

4 Relationship between Topological and Preordonnance Equivalences

4.1 Theoretical Results

We have found some theoretical results that establish a relationship between topological and preordonnance approaches. For example, from Theorem 1 of preordonnance equivalence we can deduce the following property, which states that in the case where f is strictly monotonic then if the preorder is preserved this implies that the topology is preserved and vice versa. This property can be formulated as follows:

Property 1. Let f be a strictly monotonic function of R^+ in R^+, u_i and u_j two proximity measures such that: $u_i(x,y) \rightarrow f(u_i(x,y)) = u_j(x,y)$ then,
$u_i(x,y) \leq max(u_i(x,z)\,,\, u_i(y,z)) \Leftrightarrow u_j(x,y) \leq max(u_j(x,z)\,,\, u_j(y,z))$.

Proof. Let us assume that $max(u_i(x,z)\,,\, u_i(y,z)) = u_i(x,z)$,
by Theorem 1, we provide $u_i(x,y) \leq u_i(x,z) \Rightarrow f(u_i(x,y)) \leq f(u_i(x,z))$,
again, $u_i(y,z) \leq u_i(x,z) \Rightarrow f(u_i(y,z)) \leq f(u_i(x,z))$
$\Rightarrow f(u_i(x,z)) \leq max(f(u_i(x,z)), f(u_i(y,z)))$,
hence the result, $u_j(x,y) \leq max(u_j(x,z), u_j(y,z))$.

The reciprocal implication is true, because if f is continuous and strictly monotonic then its inverse f^{-1} is continuous in the same direction of variation as f. □

Proposition 1. *In the context of topological structures induced by the relative neighbors graph, if two proximity measures u_i and u_j are equivalent in preordonnance, they are necessarily topologically equivalent.*

Proof. If $u_i \equiv u_j$ (preordonnance equivalence) then,
$u_i(x,y) \leq u_i(z,t) \Rightarrow u_j(x,y) \leq u_j(z,t) \;\; \forall x,y,z,t \in R^p$.
We have, especially for $t = x = y$ and $z \neq t$,
$u_i(x,y) \leq u_i(z,x) \Rightarrow u_j(x,y) \leq u_j(z,x)$
$u_i(x,y) \leq u_i(z,y) \Rightarrow u_j(x,y) \leq u_j(z,y)$
we deduce, $u_i(x,y) \leq max(u_i(z,x), u_i(z,y)) \Rightarrow u_j(x,y) \leq max(u_j(z,x), u_j(z,y))$
using symmetry property $P1$,
$u_i(x,y) \leq max(u_i(x,z), u_i(y,z)) \Rightarrow u_j(x,y) \leq max(u_j(x,z), u_j(y,z))$
hence, $u_i \equiv u_j$ (topological equivalence). □

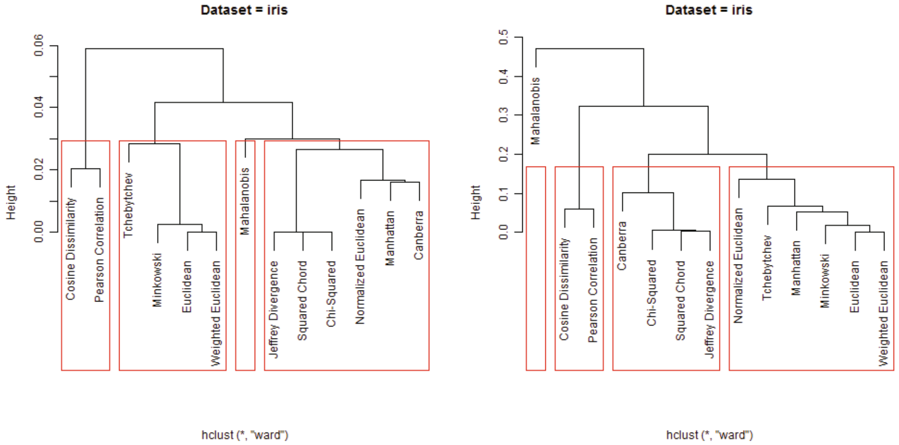
It is easy to show the following theorem from the proof of property 1.

Theorem 2. ***Equivalence in topology.*** *Let u_i and u_j be two proximity measures, if there is a strictly monotonic function f such that for every pair of objects (x,y) we have: $u_i(x,y) = f(u_j(x,y))$ then, u_i and u_j induce identical topological graphs and therefore they are equivalent.*

The converse is also true, i.e. two proximity measures which are dependent on each other induce the same topology and are therefore equivalent.

4.2 Empirical Comparisons

Comparison of Proximity Measures. We want to visualize the similarities between the proximity measures in order to see which measures are close to one another. As we already have a similarity matrix between proximity measures, we can use any classic visualization techniques to achieve this. For example, we can build a dendrogram of hierarchical clustering of the proximity measures. We can also use Multidimensional scaling or any other technique such as Laplacian projection to map the 13 proximity measures into a two dimensional space. As an illustration we show (Figure 4) the results of the Hierarchical Clustering Algorithm, HCA, on the iris dataset according to the two similarity matrices (Table 2 and Table 3) associated with each approach.



a) Topological structure: (RNG)

b) Pre-ordonnance

Fig. 4. Comparison of hierarchical trees

Now the user has two approaches, topological and preordonnance, to assess the closeness between proximity measures relative to a given dataset. This assessment might be helpful for choosing suitable proximity measures for a specific problem. Of course, there are still many questions. For instance, does the clustering of proximity measures remain identical when the data set changes? What is the sensitivity of the empirical results when we vary the number of variables or samples within the same dataset? To answer these questions we carried out a series of experiments. The core idea of these experiments was to study whether proximity measures are clustered in the same way regardless of the dataset used. To this end, given a dataset with *N* individuals and *P* variables, we verified the effect of varying the sample size, *N*, and dimension, *P*, within a given dataset and using different datasets. All datasets in our experiments were taken from the UCI repository, [24], as shown in Table 4.

Table 4. Datasets used in our experiments

Dataset Id	Name	Dimension
1	Breast Tissue	106 × 9
2	Connectionist Bench	208 × 60
3	Iris	150 × 4
4	Libras movement	360 × 91
5	Parkinsons	195 × 23
6	Waveform Database Generator (Version 2)	5000 × 40
7	Wine	178 × 13
8	Yeast	1484 × 8

- Sensitivity to change in dimension: To examine the effect of changing the dimension within a given dataset, the wave form data setwas used. 4 samples were generated by taking 10, 20, 30, and 40 variables from the dataset with 2000 individuals for the topological approach and 200 samples for the preorder approach. The results given in Tables 5 and 6 respectively show that there was a slight change in the clustering but that we could observe some stability.

- Sensitivity to change in sample size: To examine the influence of changing the number of individuals, we generated five samples from the waveform dataset varying the sample size from 1000 to 5000 for the topological approach and 100 to 400 for the preorder approach because of the complexity of the algorithm. The number of variables, 40, was the same for all experiments. The results of HCA clustering using each approach are shown in Tables 7 and 8 respectively. Clearly, there was a slight change in the clustering but it seems there was a relative stability.

- Sensitivity to varying data sets: To examine the effect of changing the data sets, the two approaches were tested with various datasets. The results are shown in Tables 9 and 10. In the topological approach, regularity {chSqr, SC, JD} and {Euc, EucW, Min} was observed regardless of the change in individuals and variables within the same dataset or across different datasets.

Table 5. The influence of varying number of variables in a data set, topological

Expt	Data set	Cluster of Proximity Measures
1	wave form[2000, 10]	{Tch}, {Man, Can}, {Cos, Pir}, {chSqr, Sc, JD, Euc, EucW, Min, NEuc, Mah}
2	wave form[2000, 20]	{Tch}, {Man, Can}, {chSqr, Sc, JD, NEuc}, {Euc, EucW, Min, Cos, Pir, Mah}
3	wave form[2000, 30]	{Tch}, {Man, Can}, {chSqr, Sc, JD, NEuc, Mah}, {Euc, EucW, Min, Cos, Pir}
4	wave form[2000, 40]	{Tch}, {Man, Can}, {chSqr, Sc, JD, NEuc, Mah}, {Euc, EucW, Min, Cos, Pir}

Table 6. The influence of varying number of variables in a data set, preorder

Expt	Data set	Cluster of Proximity Measures
1	wave form[200, 10]	{Cos, Pir}, {Mah, NEuc}, {chSqr, Sc, JD, Man, Can}, {Tch, Min, Euc, EucW}
2	wave form[200, 20]	{Tch}, {Mah}, {chSqr, Sc, JD, NEuc, Man, Can}, {Pir, Cos, Min, Euc, EucW}
3	wave form[200, 30]	{Tch}, {Mah}, {chSqr, Sc, JD, NEuc, Man, Can}, {Pir, Cos, Min, Euc, EucW}
4	wave form[200, 40]	{Tch}, {Mah}, {ChSqr, Sc, JD, NEuc, Man, Can}, {Pir, Cos, Min, Euc, EucW}

Table 7. The influence of varying size of individuals in a data set, topological

Expt	Data set	Cluster of Proximity Measures
1	wave form[1000, 40]	{Tch}, {Mah}, {Man, Can}, {Euc, EucW, Cos, Min, Pir, ChSqr, Sc, JD, NEuc}
2	wave form[2000, 40]	{Tch}, {Mah, Man, Can}, {Euc, EucW, Cos, Min, Pir}, {ChSqr, Sc, JD, NEuc}
3	wave form[3000, 40]	{Tch}, {Man, Can}, {Euc, EucW, Cos, Min, Pir}, {ChSqr, Sc, JD, NEuc, Mah}
4	wave form[4000, 40]	{Tch}, {Man, Can}, {Euc, EucW, Cos, Min, Pir}, {ChSqr, Sc, JD, NEuc, Mah}
5	wave form[5000, 40]	{Tch}, {Man, Can}, {Euc, EucW, Cos, Min, Pir}, {ChSqr, Sc, JD, NEuc, Mah}

Table 8. The influence of varying size of individuals in a data set, preorder

Expt	Data set	Cluster of Proximity Measures
1	wave form[100, 40]	{Tch}, {Mah}, {Pir, Min, Cos, Euc, EucW}, {ChSqr, Sc, JD, NEuc, Man, Can}
2	wave form[200, 40]	{Tch}, {Mah}, {Pir, Min, Cos, Euc, EucW}, {ChSqr, Sc, JD, NEuc, Man, Can}
3	wave form[300, 40]	{Can}, {Man, Tch, Pir}, {Euc, EucW, Cos, Min}, {ChSqr, Sc, JD, NEuc, Mah}
5	wave form[400, 40]	{Min, ChSqr}, {Man, JD, Mah, Sc, NEuc}, {Cos, Pir}, {Tch, Can, Euc, EucW}

Table 9. The influence of varying datasets, topological

Expt	Data set	Cluster of Proximity Measures
1	Iris [150, 4]	{Pir, Cos}, {Mah}, {Euc, EucW, Min, Tch} , {chSqr, Sc, JD, NEuc, Man, Can}
2	Breast Tissue[106, 9]	{Sc, JD}, {Euc, EucW, Min, Tch, Man, chSqr}, {Cos, Pir}, {Mah, Can, NEuc}
3	Parkinsons [195, 23]	{chSqr, Sc, JD }, {Euc, EucW, Min, Man, Tch}, {Pir, Cos}, {NEuc, Can, Mah}
4	C.Bench [208, 60]	{chSqr, Sc, JD }, {Tch}, {Can, NEuc}, {Euc, EucW, Min, Cos, Pir, Man, Mah}
5	Wine [178, 13]	{chSqr, Sc, JD}, {Euc, EucW, Min, Man, Tch}, {Cos, Pir}, {Mah, Can, NEuc}
6	Yeast [1484, 8]	{chSqr}, {JD}, {Tch}, {Cos, Pir, Sc, Euc, EucW, Min, Mah, NEuc, Man, Can}
7	L.Movement [360, 91]	{JD}, {Mah}, {Cos, Pir, Tch}, {Euc, EucW, Min, NEuc, chSqr, Sc, Man, Can}
8	wave form[5000, 40]	{Tch}, {Man, Can}, {Euc, EucW, Cos, Min, Pir}, {ChSqr, Sc, JD, NEuc, Mah}

Table 10. The influence of varying datasets, preorder

Expt	Data set	Cluster of Proximity Measures
1	Iris [150, 4]	{Mah}, {Cos, Pir}, {Euc, EucW, Min, Man, Tch, NEuc} , {Can, chSqr, Sc, JD}
2	Breast Tissue[106, 9]	{Cos, Pir}, {Sc, JD}, {Mah, Can, NEuc}, {Euc, EucW, Min, Tch, Man, chSqr}
3	Parkinsons [195, 23]	{Mah}, {Can, NEuc}, {Cos, Pir}, {chSqr, Sc, JD ,Euc, EucW, Min, Man, Tch}
4	Wine [178, 13]	{Mah}, {Can, NEuc}, {Cos, Pir}, {chSqr, Sc, JD, Euc, EucW, Min, Man, Tch}
5	L.Movement [360, 91]	{Can, NEuc, WEuc, Euc, Pir}, {Man, Min}, {Mah, Tch, Sc}, {JD, Cos, ChSqr}

5 Conclusion

In this paper, we have proposed a new approach for comparing proximity measures with complexity $O(n^2)$. This approach produces results that are not totally identical to those produced by former methods. One might wonder which approach is the best. We believe that this question is not relevant. The topological approach described here has some connections with preordonnance, but proposes another point of view for comparison. The topological approach has a lower time complexity. From theoretical analysis, when a proximity measure is a function of another proximity measure then we have shown that the two proximity measures are identical for both approaches. When this is not the case, the experimental analysis showed that there is sensitivity to sample size, dimensionality and the dataset used.

References

1. Batagelj, V., Bren, M.: Comparing resemblance measures. In: Proc. International Meeting on Distance Analysis, DISTANCIA 1992 (1992)
2. Batagelj, V., Bren, M.: Comparing resemblance measures. Journal of classification 12, 73–90 (1995)
3. Bouchon-Meunier, M., Rifqi, B., Bothorel, S.: Towards general measures of comparison of objects. Fuzzy Sets and Systems 84(2), 143–153 (1996)
4. Clarke, K.R., Somerfield, P.J., Chapman, M.G.: On resemblance measures for ecological studies, including taxonomic dissimilarities and a zero-adjusted Bray-Curtis coefficient for denuded assemblages. Journal of Experimental Marine Biology & Ecology 330(1), 55–80 (2006)
5. Fagin, R., Kumar, R., Sivakumar, D.: Comparing top k lists. In: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics (2003)
6. Kim, J.H., Lee, S.: Tail bound for the minimal spanning tree of a complete graph. Statistics & Probability Letters 64(4), 425–430 (2003)
7. Lerman, I.C.: Indice de similarité et préordonnance associée, Ordres. In: Travaux Du Séminaire Sur Les Ordres Totaux Finis, Aix-en-Provence (1967)
8. Lesot, M.J., Rifqi, M., Benhadda, H.: Similarity measures for binary and numerical data: a survey. IJKESDP 1(1), 63–84 (2009)
9. Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the 15th International Conference on Machine Learning, pp. 296–304 (1998)
10. Liu, H., Song, D., Ruger, S., Hu, R., Uren, V.: Comparing Dissimilarity Measures for Content-Based Image Retrieval. In: Li, H., Liu, T., Ma, W.-Y., Sakai, T., Wong, K.-F., Zhou, G. (eds.) AIRS 2008. LNCS, vol. 4993, pp. 44–50. Springer, Heidelberg (2008)
11. Malerba, D., Esposito, F., Gioviale, V., Tamma, V.: Comparing dissimilarity measures for symbolic data analysis. In: Proceedings of Exchange of Technology and Know-how and New Techniques and Technologies for Statistics, vol. 1, pp. 473–481 (2001)
12. Malerba, D., Esposito, F., Monopoli, M.: Comparing dissimilarity measures for probabilistic symbolic objects. In: Data Mining III. Series Management Information Systems, vol. 6, pp. 31–40 (2002)
13. Mantel, N.: A technique of disease clustering and a generalized regression approach. Cancer Research 27, 209–220 (1967)
14. Noreault, T., McGill, M., Koll, M.B.: A performance evaluation of similarity measures, document term weighting schemes and representations in a Boolean environment. In: Proceedings of the 3rd Annual ACM Conference on Research and Development in Information Retrieval (1980)
15. Park, J.C., Shin, H., Choi, B.K.: Elliptic Gabriel graph for finding neighbors in a point set and its application to normal vector estimation. Computer-Aided Design 38(6), 619–626 (2006)
16. Preparata, F.P., Shamos, M.I.: Computational geometry: an introduction. Springer (1985)

17. Richter, M.M.: Classification and learning of similarity measures. In: Proceedings der Jahrestagung der Gesellschaft fur Klassifikation. Studies in Classification, Data Analysis and Knowledge Organisation. Springer (1992)
18. Rifqi, M., Detyniecki, M., Bouchon-Meunier, B.: Discrimination power of measures of resemblance. In: IFSA 2003. Citeseer (2003)
19. Schneider, J.W., Borlund, P.: Matrix comparison, Part 1: Motivation and important issues for measuring the resemblance between proximity measures or ordination results. Journal of the American Society for Information Science and Technology 58(11), 1586–1595 (2007)
20. Schneider, J.W., Borlund, P.: Matrix comparison, Part 2: Measuring the resemblance between proximity measures or ordination results by use of the Mantel and Procrustes statistics. Journal of the American Society for Information Science and Technology 58(11), 1596–1609 (2007)
21. Spertus, E., Sahami, M., Buyukkokten, O.: Evaluating similarity measures: a large-scale study in the orkut social network. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. ACM (2005)
22. Strehl, A., Ghosh, J., Mooney, R.: Impact of similarity measures on web-page clustering. In: Workshop on Artificial Intelligence for Web Search, pp. 58–64. AAAI (2000)
23. Toussaint, G.T.: The relative neighbourhood graph of a finite planar set. Pattern Recognition 12(4), 261–268 (1980)
24. UCI Machine Learning Repository, http://archive.ics.uci.edu/ml
25. Ward, J.R.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association, JSTOR 58(301), 236–244 (1963)
26. Warrens, M.J.: Bounds of resemblance measures for binary (presence/absence) variables. Journal of Classification 25(2), 195–208 (2008)
27. Zhang, B., Srihari, S.N.: Properties of binary vector dissimilarity measures. In: Proc. JCIS Int'l Conf. Computer Vision, Pattern Recognition, and Image Processing, vol. 1 (2003)
28. Zwick, R., Carlstein, E., Budescu, D.V.: Measures of similarity among fuzzy concepts: A comparative analysis. Int. J. Approx. Reason 2(1), 221–242 (1987)

Quad-tuple PLSA: Incorporating Entity and Its Rating in Aspect Identification

Wenjuan Luo[1,2], Fuzhen Zhuang[1], Qing He[1], and Zhongzhi Shi[1]

[1] The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[2] Graduate University of Chinese Academy of Sciences, Beijing 100039, China
{luowj,zhuangfz,heq,shizz}@ics.ict.ac.cn

Abstract. With the opinion explosion on Web, there are growing research interests in opinion mining. In this study we focus on an important problem in opinion mining — Aspect Identification (AI), which aims to extract aspect terms in entity reviews. Previous PLSA based AI methods exploit the 2-tuples (e.g. the co-occurrence of head and modifier), where each latent topic corresponds to an aspect. Here, we notice that each review is also accompanied by an entity and its overall rating, resulting in quad-tuples joined with the previously mentioned 2-tuples. Believing that the quad-tuples contain more co-occurrence information and thus provide more ability in differentiating topics, we propose a model of Quad-tuple PLSA, which incorporates two more items — entity and its rating, into topic modeling for more accurate aspect identification. The experiments on different numbers of hotel and restaurant reviews show the consistent and significant improvements of the proposed model compared to the 2-tuple PLSA based methods.

Keywords: Quad-tuple PLSA, Aspect Identification, Opinion Mining.

1 Introduction

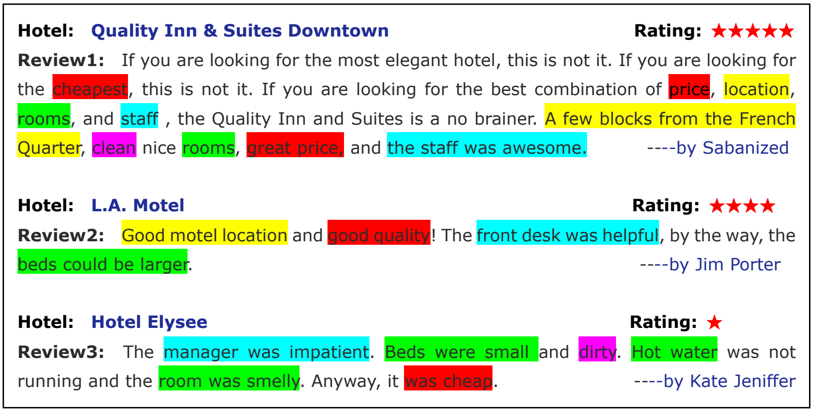
With the Web 2.0 technology encouraging more and more people to participate in online comments, recent years have witnessed the opinion explosion on Web. As large scale of user comments accumulate, it challenges both the merchants and customers to analyze the opinions or make further decisions. As a result, opinion mining which aims at determining the sentiments of opinions has become a hot research topic.

Additionally, besides the simple overall evaluation and summary, both customers and merchants are becoming increasingly concerned in certain aspects of the entities. Take a set of restaurant reviews as example. Common restaurant aspects include “food”, “service”, “value” and so on. Some guests may be interested in the “food” aspect, while some may think highly of the “value” or “service” aspect. To meet these personalized demands, we need to decompose the opinions into different aspects for better understanding or comparison.

On the other hand, it also brings out perplexity for merchants to digest all the customer reviews in case that they want to know in which aspect they

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 392–404, 2012.

lack behind their competitors. As pointed out in [12], the task of aspect-based summarization consists of two subtasks: the first is Aspect Identification (AI), and the second is sentiment classification and summarization. The study in this paper mainly focuses on the first task, which aims to accurately identify the aspect terms in the reviews for certain type of entities.



Hotel: Quality Inn & Suites Downtown **Rating: ★★★★★**

Review1: If you are looking for the most elegant hotel, this is not it. If you are looking for the cheapest, this is not it. If you are looking for the best combination of price, location, rooms, and staff , the Quality Inn and Suites is a no brainer. A few blocks from the French Quarter, clean nice rooms, great price, and the staff was awesome. ----by Sabanized

Hotel: L.A. Motel **Rating: ★★★★**

Review2: Good motel location and good quality! The front desk was helpful, by the way, the beds could be larger. ----by Jim Porter

Hotel: Hotel Elysee **Rating: ★**

Review3: The manager was impatient. Beds were small and dirty. Hot water was not running and the room was smelly. Anyway, it was cheap. ----by Kate Jeniffer

Fig. 1. Sample Reviews

As shown in Figure 1, there are 3 reviews on different hotels, where the description for the same aspect is stained in the same color. One of a recent works in this area argues that it is more sensible to extract aspects from the phrase level rather than the sentence level since a single sentence may cover different aspects of an entity (as shown in Figure 1, a sentence may contain different colored terms) [5]. Thus, Lu et al. decompose reviews into phrases in the form of (*head*, *modifier*) pairs. A head term usually indicates the aspect while a modifier term reflects the sentiment towards the aspect. Take the phrase "excellent staff" for example. The head "staff" belongs to the "staff/front desk" aspect, while the modifier "excellent" shows a positive attitude to it. Utilizing the (*head*, *modifier*) pairs, they explore the latent topics embedded in it with aspect priors. In other words, they take the these 2-tuples as input, and output the latent topics as the identified aspects.

In this study, we observe that besides the *(head, modifier)* pairs each review is often tied with an entity and its overall rating. As shown in Figure 1, a hotel name and an overall rating are given for each review. Thus, we can construct the quad-tuples of

$$(head,\ modifier,\ rating,\ entity),$$

which indicates that a phrase of the *head* and *modifier* appears in the review for this *entity* with the *rating*. For example, the reviews in Figure 1 include the following quad-tuples,

(*price*, *good*, *5*, *Quality Inn*); (*staff*, *awesome*, *5*, *Quality Inn*);
(*location*, *good*, *4*, *L.A.Motel*); (*bed*, *small*, *1*, *Hotel Elysee*).

With these quad-tuples from the reviews for a certain type of entities, we further argue that they contain more co-occurrence information than 2-tuples, thus provide more ability in differentiating terms. For example, reviews with the same rating tend to share similar modifiers. Additionally, reviews with the same rating on the same entity often talk about the same aspects of that entity (imagine that people may always assign lowest ratings to an entity because of its low quality in certain aspect). Therefore, incorporating entity and rating into the tuples may facilitate aspect generation.

Motivated by this observation, we propose a model of Quad-tuple PLSA (QPLSA for short), which can handle two more items (compared to the previous 2-tuple PLSA [1,5]) in topic modeling. In this way we aim to achieve higher accuracy in aspect identification. The rest of this paper is organized as follows: Section 2 presents the problem definition and preliminary knowledge. Section 3 details our model Quad-tuple PLSA and the EM solution. Section 4 gives the experimental results to validate the superiority of our model. Section 5 discusses the related work and we conclude our paper in Section 6.

2 Problem Definition and Preliminary Knowledge

In this section, we first introduce the problem, and then briefly review Lu's solution–the Structured Probabilistic Latent Semantic Analysis (SPLSA) [5]. The frequently used notations are summarized in Table 1.

Table 1. Frequently used notations

Symbol	Description
t	the comment
$\mathbf{T}$	the set of comments
h	the head term
m	the modifier term
e	the entity
r	the rating of the comment
q	the quad-tuple of (h,m,r,e)
z	the latent topic or aspect
K	the number of latent topics
Λ	the parameters to be estimated
$n(h, m)$	the number of co-occurrences of head and modifier
$n(h, m, r, e)$	the number of co-occurrences of head,modifier, rating and entity
$\mathbf{X}$	the whole data set

2.1 Problem Definition

In this section, we give the problem definition and the related concepts.

Definition 1 (Phrase). *A phrase $f = (h, m)$ is in the form of a pair of head term h and modifier m. And SPLSA adopts such (head, modifier) 2-tuple phrases for aspect extraction.*

Definition 2 (Quad-tuple). *A quad-tuple $q = (h, m, r, e)$ is a vector of head term h, modifier m, rating r and entity e. Given a review on entity e with rating r, we can generate a set of quad-tuples, denoted by*

$$\{(h, m, r, e)|Phrase\,(h, m) \;\; appears \;\; with \;\; rating \;\; r \;\; in \;\; a \;\; review \;\; of \;\; entity \;\; e\}.$$

Aspect Cluster. An aspect cluster A_i is a cluster of head terms which share similar meaning in the given context. We represent $A_i = \{h|\mathcal{G}(h) = i\}$, where $\mathcal{G}$ is a mapping function that maps h to a cluster aspect A_i.

Aspect Identification. The goal of aspect identification is to find the mapping function $\mathcal{G}$ that correctly assigns the aspect label for given head term h.

2.2 Structured PLSA

Structured PLSA (SPLSA for short) is a 2-tuple PLSA based method for rated aspect summarization. It incorporates the structure of phrases into the PLSA model, using the co-occurrence information of head terms and their modifiers. Given the whole data $\mathbf{X}$ composed of (head, modifier) pairs, SPLSA arouses a mixture model with latent model topics z as follows,

$$p(h, m) = \sum_{z} p(h|z)p(z|m)p(m). \tag{1}$$

The parameters of $p(z|m)$, $p(h|z)$ and $p(m)$ can be obtained using the EM algorithm by solving the maximum log likelihood problem in the following,

$$\log p(\mathbf{X}|\Lambda) = \sum_{h,m} n(h, m) \log \sum_{z} p(z|m)p(h|z)p(m), \tag{2}$$

where Λ denotes all the parameters. And the prior knowledge of seed words indicating specific aspect are injected in the way as follows:

$$p(h|z; \Lambda) = \frac{\sum_{m} n(h, m)p(z|h, m; \Lambda^{old}) + \sigma p(h|z_0)}{\sum_{h'} \sum_{m} n(h', m)p(z|h', m; \Lambda^{old}) + \sigma}, \tag{3}$$

where z_0 denotes the priors corresponding to the latent topic z, and σ is the confidential parameter of the head term h belonging to aspect z_0. And each h is grouped into topic z with the largest probability of generating h, which was the aspect identification function in SPLSA: $A(h) = \arg\max_z p(h|z)$.

3 QPLSA and EM Solution

3.1 QPLSA

In SPLSA, aspects are extracted based on the co-occurrences of head and modifier, namely a set of 2-tuples. Next, we will detail our model–QPLSA, which takes the quad-tuples as input for more accurate aspect identification.

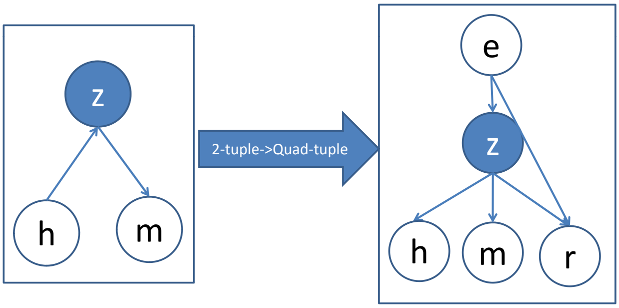


Fig. 2. From SPLSA Model to QPLSA Model

Figure 2 illustrates the graphical model of QPLSA. The directed lines among the nodes are decided by the understandings on the dependency relationships among these variables. Specifically, we assume that given a latent topic z, h and m are conditionally independent. Also, a reviewer may show different judgement toward different aspects of the same entity. Thus, rating r is jointly dependent on entity e and latent topic z. From the graphic model in Figure 2, we can write the joint probability over all variables as follows:

$$p(h,m,r,e,z) = p(m|z)p(h|z)p(r|z,e)p(z|e)p(e). \tag{4}$$

Let $\mathbf{Z}$ denote all the latent variables, and given the whole data $\mathbf{X}$, all the parameters can be approximated by maximizing the following log likelihood function,

$$\log p(\mathbf{X}|\Lambda) = \log \sum_{\mathbf{Z}} p(\mathbf{X},\mathbf{Z}|\Lambda) \quad = \sum_{h,m,r,e} n(h,m,r,e) \log \sum_{z} p(h,m,r,e,z|\Lambda), \tag{5}$$

where Λ includes the parameters of $p(m|z)$, $p(h|z)$, $p(r|z,e)$, $p(z|e)$ and $p(e)$. The derivation of EM algorithm is detailed in next subsection.

3.2 Deriving the EM Solution

Traditionally, the Expectation-Maximization(EM) algorithm is utilized for optimization of PLSA based methods. In our model, we also adopt the EM algorithm to maximize the log likelihood function in Equation (5). Specifically, the lower bound (Jensen's inequality) $\mathcal{L}_0$ of (5) is:

$$\mathcal{L}_0 = \sum_z q(z) \log\{\frac{p(h,m,r,e,z|\Lambda)}{q(z)}\}. \tag{6}$$

where $q(z)$ could be an arbitrary function, and here we set $q(z) = p(z|h,m,r,e;\Lambda^{old})$ and substitute into (6):

$$\begin{aligned}\mathcal{L}_0 = &\underbrace{\sum_z p(z|h,m,r,e;\Lambda^{old}) \log p(z,h,m,r,e|\Lambda)}_{\mathcal{L}} \\ &\underbrace{-\sum_z p(z|h,m,r,e;\Lambda^{old}) \log\{p(z|h,m,r,e;\Lambda^{old})\}}_{const} = \quad \mathcal{L} + const.\end{aligned} \tag{7}$$

E Step: Constructing $\mathcal{L}$. For the solution of (5),we have:

$$\mathcal{L} = \sum_{h,m,r,e,z} n(h,m,r,e)p(z|h,m,r,e;\Lambda^{old}) \cdot \log[p(e)p(z|e)p(h|z)p(m|z)p(r|e,z)], \tag{8}$$

where

$$p(z|e,h,m,r) = \frac{p(e)p(z|e)p(h|z)p(m|z)p(r|e,z)}{\sum_z p(e)p(z|e)p(h|z)p(m|z)p(r|e,z)}. \tag{9}$$

M Step: Maximizing $\mathcal{L}$. Here we maximize $\mathcal{L}$ with its parameters by Lagrangian Multiplier method. Expand $\mathcal{L}$ and extract the terms containing $p(h|z)$. Then, we have $\mathcal{L}_{[p(h|z)]}$ and apply the constraint $\sum_h p(h|z) = 1$ into the following equation:

$$\frac{\partial[\mathcal{L}_{[p(h|z)]} + \lambda(\sum_h p(h|z) - 1)]}{\partial p(h|z)} = 0, \tag{10}$$

we have

$$\hat{p}(h|z) \propto \sum_{m,r,e} p(z|h,m,r,e;\Lambda^{old}). \tag{11}$$

Note that $\hat{p}(h|z)$ should be normalized via

$$\hat{p}(h|z) = \frac{\sum_{m,r,e} n(h,m,r,e)p(z|h,m,r,e;\Lambda^{old})}{\sum_{h',m,r,e} n(h',m,r,e)p(z|h',m,r,e;\Lambda^{old})}. \tag{12}$$

Similarly, we have:

$$p(e) = \frac{\sum_{z,h,m,r} n(h,m,r,e)p(z|e,h,m,r;\Lambda^{old})}{\sum_{h,m,r,e} n(h,m,r,e;\Lambda^{old})}, \tag{13}$$

$$p(z|e) = \frac{\sum_{h,m,r} n(h,m,r,e)p(z|e,h,m,r;\Lambda^{old})}{\sum_{h,m,r,z'} n(h,m,r,e)p(z'|e,h,m,r;\Lambda^{old})}, \tag{14}$$

$$p(m|z) = \frac{\sum_{e,h,r} n(h,m,r,e)p(z|e,h,m,r;\Lambda^{old})}{\sum_{e,h,r,m'} n(h,m',r,e)p(z|e,h,m',r;\Lambda^{old})}, \tag{15}$$

$$p(r|z,e) = \frac{\sum_{h,m} n(h,m,r,e)p(z|e,h,m,r;\Lambda^{old})}{\sum_{h,m,r'} n(h,m,r',e)p(z|e,h,m,r';\Lambda^{old})}. \tag{16}$$

3.3 Incorporating Aspect Prior

For specific aspect identification, we may have some domain knowledge about aspects. For instance, the aspect "food" may include a few seed words such as "breakfast", "potato", "drink" and so on. Specifically, we use a unigram language model $p(h|z)$ to inject the prior knowledge for the aspect z. Take the aspect "food" as an example, we can assign the conditional probability $p(\text{breakfast}|\text{food})$, $p(\text{potato}|\text{food})$ and $p(\text{drink}|\text{food})$ with a high value of probability τ (e.g., $\tau(0 \leq \tau \leq 1)$ is a pre-defined threshold).

Similarly with the method in Lu et al. [5], we introduce a conjugate Dirichlet prior on each unigram language model, parameterized as $Dir(\sigma p(h|z) + 1)$, and σ denotes the confidence for the prior knowledge of aspect z. Specifically, the prior for all the parameters is given by:

$$p(\Lambda) \propto \prod_z \prod_h p(h|z)^{\sigma p(h|z)} \tag{17}$$

where $\sigma = 0$ if we have no prior knowledge on z. Note that adding the prior can be interpreted as increasing the counts for head term h by $\sigma + 1$ times when estimating $p(h|z)$. Therefore, we have:

$$p(h|z;\Lambda) \quad = \frac{\sum_{m,r,e} n(h,m,r,e)p(z|h,m,r,e;\Lambda^{old}) + \sigma p(h|z)}{\sum_{h',m,r,e} n(h',m,r,e)p(z|h',m,r,e;\Lambda^{old}) + \sigma}. \tag{18}$$

3.4 Aspect Identification

Our goal is to assign the head term h to a correct aspect label, and we follow the mapping function $\mathcal{G}$ as SPLSA [5]:

$$\mathcal{G}(h) = \arg\max_z p(h|z), \tag{19}$$

where we select the aspect which generates h with the largest probabilty as the aspect label for head term h.

4 Experiments

In this section, we present the experimental results to evaluate our model QPLSA. Firstly, we introduce the data sets and implementation details, and then give the experimental results in the following subsections.

4.1 Data Sets

We adopt two different datasets for evaluation, which are detailed in Table 2. The first dataset is a corpus of hotel reviews provided by Wang et al. [14]. The data set includes 246,399 reviews on 1850 hotels with each review associated with an overall rating and 7 detailed ratings about the pre-defined aspects, and the value of the rating ranges from 1 star to 5 stars. Table 2 also lists the prior knowledge of some seed words indicating specific aspects.

The other dataset is about restaurant reviews from Snyder et al. [11], which is much sparser than the previous one. This dataset contains 1609 reviews on 420 restaurants with each review associated with an overall rating and 4 aspect ratings. For both of the datasets, we decompose the reviews into phrases utilizing a set of NLP toolkits such as the POS tagging and chunking functions[1].

4.2 Implementation Details

terms and manually label them as knowledge base. Specifically, for the hotel reviews we select 408 head terms and categorize them into 7 specific aspects. While for the restaurant reviews, we select 172 head terms and label them with 4 specific aspects. The details of the categorization are summarized in Table 3, and A1 to A7 corresponds to the aspects in Table 2. Here we only evaluate the results of specific aspect identification and compare our model QPLSA with SPLSA.

Table 2. Pre-defined Aspects and Prior Knowledge

Hotel Reviews		
Aspects	Prior Words	Aspect No.
Value	value,price,quality,worth	A1
Room	room,suite,view,bed	A2
Location	location,traffic,minute,restaurant	A3
Cleanliness	clean,dirty,maintain,smell	A4
Front Desk/Staff	staff,check,help,reservation	A5
Service	service,food,breakfast,buffet	A6
Business	business,center,computer,internet	A7
Restaurant Reviews		
Food	food,breakfast,potato,drink	A1
Ambience	ambience,atmosphere,room,seat	A2
Service	service,menu,staff,help	A3
Value	value,price,quality,money	A4

[1] http://opennlp.sourceforge.net/

Table 3. Aspect Identification Accuracy on Two Datasets

	Hotel Reviews								Restaurant Reviews					All
	A1	A2	A3	A4	A5	A6	A7	A1-7	A1	A2	A3	A4	A1-4	All
Categorized	52	108	93	35	39	64	17	408	73	32	42	25	172	580
QPLSA	29	69	45	21	31	47	12	**254**	29	21	23	22	**95**	**349**
SPLSA	29	61	46	20	28	46	4	234	4	0	7	5	16	250
Q-accuracy	0.56	0.64	0.48	0.60	0.79	0.73	0.71	**0.62**	0.39	0.66	0.55	0.88	**0.55**	**0.60**
S-accuracy	0.56	0.56	0.49	0.57	0.72	0.72	0.24	0.57	0.05	0	0.17	0.2	0.09	0.43

4.3 Experimental Results

Aspect Identification. We present the accuracy of aspect identification of all the head terms in Table 3. Since we focus on specific aspect extraction, our discussions only detail the results on specific aspects. In the table, Ai denote the i-th specific aspect as described in Table 2, and "A1-7" and "A1-4" denote the sum of the specific aspects for hotel reviews and restaurant reviews, respectively.

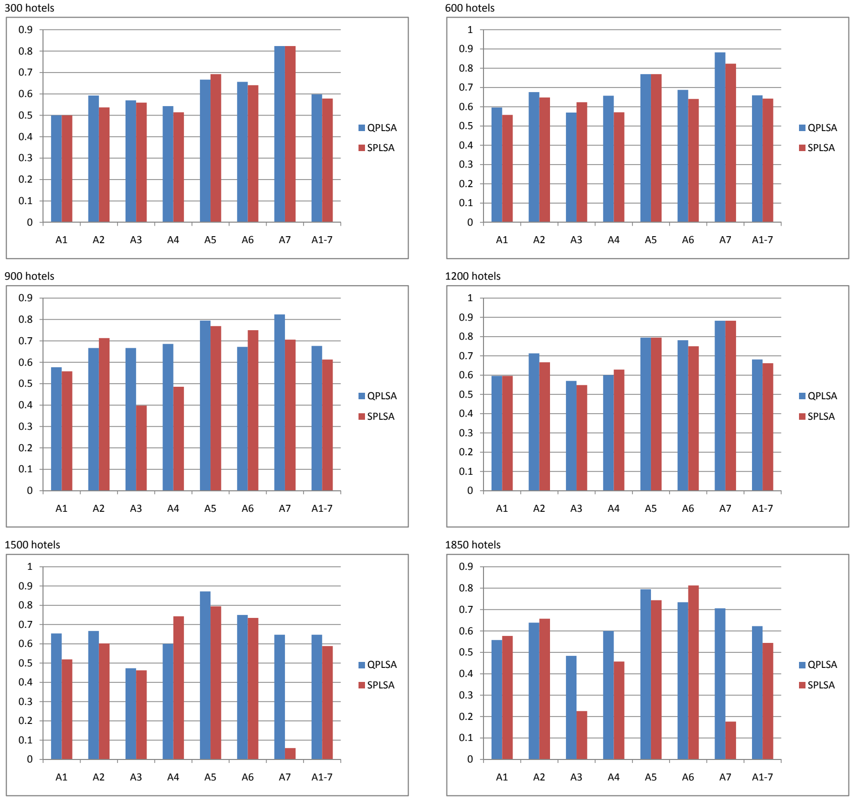


Fig. 3. Accuracy on different numbers of hotels

In Table 3, Q-accuracy denotes the accuracy of QPLSA, and S-accuracy represents that of SPLSA. From the results reported in Table 3, apparently, QPLSA achieves better performance compared to SPLSA. As can be seen, the accuracy of QPLSA for all the reviews is much higher than that of SPLSA, which indicates that quad-tuples exploits more information for specific aspect generation as opposed to 2-tuples. All the experimental results demonstrate the effectiveness of incorporating entity and its rating for aspect identification.

To further validate the superiority of QPLSA over SPLSA, we conduct systematic experiments on different data sets of hotel reviews for comparison. We carry out experiments on different numbers of hotels (e.g., 300, 600, 900, 1200, 1500 and 1850), and all the results are shown in Figure 3.

As illustrated in Fig. 3, in particular, the performance of QPLSA varies for different aspects due to the skrewness of corpse over specific topics. Nevertheless, for different numbers of hotels, that the overall accuracy of QPLSA always outperforms that of SPLSA strongly supports that Aspect Identification of QPLSA can benefit from the additional information of entity and its rating.

Representative Term Extraction. Table 4 lists representative terms for the 7 specific aspects of hotel reviews and the 4 aspects of the restaurant reviews. For each aspect, we choose 20 head terms with the largest probability, and the terms that are correctly associated with the aspects are marked with bold and italic.

Table 4. Representative terms for Different Aspects

Hotel Reviews		
Aspects	Representative Terms By QPLSA	Representative Terms By SPLSA
Value	hotel location experience ***value price*** size vacation ***rates choice deal*** job way surprise atmosphere ***quality selections money*** holiday variety spots	walk ***value price rates*** side york parking station tv orleans ***quality*** distance standards screen light ***money*** end ***charge*** line bus
Room	***room bed view pool bathroom suits*** ocean ***shower style space feel window facilities*** touch ***balcony chair bath amenities pillows furnished***	***room*** quarters area ***bed view pool*** transportation ***bathroom suits*** towels ***shower*** variety lobby ***space window facilities balcony chair bath*** sand
Location	***places restaurants area walk resort beach city street shopping minutes bus distance quarters building tourist store tour*** lobby attractions cafe	time ***restaurants*** day night ***resort trips beach*** doors ***street way minutes*** years week hour ***visit*** weekend ***block island*** evening morning
Cleanliness	***water decor towels*** fruit ***tub air*** appointed sand ***cleaning smell maintained noise music*** club ***condition*** garden republic done design francisco	floor level ***water*** flight ***air noise music*** class worlds ***cleaning smell maintained condition*** wall francisco car eggs anniversary ***notch*** afternoon
Front Desk	***staff reservation guests checking manager*** house ***airporter receptions desk help*** island eggs lady ***attitude smiles*** lounge museum kong man ***concierge***	***staff desk*** people ***guests checking*** person couples ***manager*** fun lounge children ***member receptions*** towers guys ***reservation*** cart trouble ***attitude*** lady
Service	***service breakfast food bar drinks buffet*** tv ***coffee meals wine bottle items dinner juice tea snacks dish*** screen car shuttle	***service breakfast food*** access ***bar*** tub shuttle ***drinks buffet coffee meals fruit wine bottle*** connected weather ***juice beer tea snacks***
Business Service	floor ***access internet*** side parking station standards light end class ***line sites*** wall stop ***business connected center*** district towers level	shopping problem building complaints ones ***internet*** traveller points bit tourist store cafe deal thing attractions issue star ***sites*** items city
Total	**89 correct terms**	64 correct terms
Restaurant Reviews		
Food	***food potato sauce ribs wine taste drinks fries*** parking fee dogs ***toast breakfast bun cajun pancakes croissants lasagna pies cinnamon***	***food potato sauce ribs wine sause taste drinks*** gravy diversity reduction ***feast charcoal*** plus brats nature ***tiramisu cauliflower goods***
Ambience	***atmosphere style*** cheese shrimp ***room seated music*** tomatoes ***decor game dressing*** tip orders onion mushroom garlic cocktail ***setting piano*** mousse	***atmosphere*** area ***style room seated feeling music manner piano band poster arts cello movie blues appearance*** folk medium francisco avenue
Service	***service staff menu wait guy guests carte*** chili ***attitude*** space downtown section become women ***employees critic*** poster market ***waitstaff office***	***help service staff menu attitude guests*** gras mousse maple behavior tone lettuce defines future excuse smorgasbord sports ***networkers*** supper grandmothers
Value	***priced value quality*** done management legs anniversary ***rate money*** thought cafeteria informed croutons bags elaine system bomb ***proportions recipes buy***	***priced value quality*** parking ***rate money*** ravioli ***fee*** pupils flaw heron inside winter education aiken standbys drenched ***paying*** year-old-home veteran
Total	**47 correct terms**	42 correct terms
All	**136 correct terms**	108 correct terms

Totally, for the 7 aspects of hotel reviews, there are 105 head terms accurately selected by QPLSA compared to 64 by SPLSA. Also for the 4 aspects of restaurant reviews, more correct words are captured by QPLSA than SPLSA. In all, QPLSA extracts 136 correct terms compared to 108 of SPLSA. All these results demonstrate that incorporating entity and its rating for aspect identification(or extraction) is effective.

Note that both QPLSA and SPLSA obtain much better results on dataset hotel reviews than those on restaurant reviews. The reason is that both methods are based on generative model that models the co-occurrence information. As we know, hotel review dataset is much more dense, and thus can provide enough co-occurrence information for learning.

5 Related Work

This section details some interesting study that is relevant to our research. Pang et al. [8] give a full overview of opinion mining and sentiment analysis, after describing the requests and challenges, they outlined a series of approaches and applications for this research domain. It is pointed out that sentiment classification could be broadly referred as binary categorization, multi-class categorization, regression or ranking problems on an opinionated document.

Hu and Liu [2] adopt association mining based techniques to find frequent features and identify the polarity of opinions based on adjective words. However, their method did not perform aspect clustering for deeper understanding of opinions. Similar work carried out by Popescu and Etzioni [10] achieved better performance on feature extraction and sentiment polarity identification, however, there is still no consideration of aspects.

Kim et al. [3] developed a system for sentiment classification through combining sentiments at word and sentence levels, however their system did not help users digest opinions from the aspect perspective. More approaches for sentiment analysis could be referred to [9,13,15,7], although none of these methods attach importance to aspects.

Topic models [14,4,6,5] are also utilized to extract aspects from online reviews. Lu et al. adopt the unstructured and structured PLSA for aspect identification [5], however, in their model, there is no consideration of rating or entity in the aspect generation phase. Wang et al. [14] proposed a rating regression approach for latent aspect rating analysis on reviews, still in their model they do not take account of entity. Mei et al. [6] defined the problem of topic-sentiment analysis on Weblogs and proposed Topic-Sentiment Mixture(TSM) model to capture sentiments and extract topic life cycles. However, as mentioned before, none of these topic models extracts aspects in view of quads.

A closely related work to our study could be referred to Titov and McDonald's [12] work on aspect generation. They construct a joint statistical model of text and sentiment ratings, called the Multi-Aspect Sentiment model(MAS) to generate topics from the sentence level. They build local and global topics based on the Multi-Grain Latent Dirichlet Allocation model (MG-LDA) for better aspect generation. One recent work [4] by Lakkaraju et al. also focused on sentence level aspect identification. However, according to our observation, a single sentence may address several different aspects and therefore we generate aspects from the phrase level, while they extract topics from the sentence level. Moreover, in their model, there is no consideration of entity.

6 Conclusion

In this paper, we focus on aspect identification in opinion mining and propose a quad-tuple PLSA based model which novelly incorporates the rating and entity for a better aspect generation. Compared to traditional 2-tuple(head, modifier) PLSA based modeling methods, our model exploits the co-occurrance information among quad-tuples(head, modifier, rating, entity) and extract aspects from a finer grain. After formally describing our quad-tuple PLSA(QPLSA) and applying the EM algorithm for optimization, we carry out systematic experiments to testify the effectiveness of our algorithm. Experimental results show that this method achieves better performance in aspect identification and representative term extraction compared to SPLSA(a 2-tuple PLSA based method). Our future work will focus on aspect rating prediction and sentiment summarization.

Acknowledgement. We would like to thank Ping Luo from the H.P.(Hewlett-Packard) Labs China. Besides, our work is supported by the National Natural Science Foundation of China (No. 60933004, 60975039, 61035003, 60903141, 61072085), National Basic Research Priorities Programme (No.2007CB311004) and National Science and Technology Support Plan (No.2006BAC08B06).

References

1. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd International Conference on Reserach and Development in Inforamtion Retrieval, SIGIR 1999 (1999)
2. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD 2004), pp. 168–177 (2004)
3. Kim, S.M., Hovy, E.: Determining the sentiment of opinors. In: Proceedings of the 20th International Conference on Computational Linguistics, p. 1367 (2004)
4. Lakkaraju, H., Bhattacharyya, C., Bhattacharya, I., Merugu, S.: Exploiting coherence for the simultaneous discovery of latent facets and associated sentiments. In: Proceedings of 2011 SIAM International Conference on Data Mining (SDM 2011), pp. 498–509 (April 2011)
5. Lu, Y., Zhai, C., Sundaresan, N.: Rated aspect summarization of short comments. In: Proceedings of the 18th International Conference on World Wide Web (WWW 2009), pp. 131–140 (2009)
6. Mei, Q., Ling, X., Wondra, M., Su, H., Zhai, C.: Topic sentiment mixture: Modeling facets and opinions in weblogs. In: Proceedings of the 16th International World Wide Web Conference (WWW 2007), pp. 171–180 (2007)

7. Morinaga, S., Tateishi, K.Y.K., Fukushima, T.: Mining product reputations on the web. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2002), pp. 341–349 (2002)
8. Pang, B., Lee, L.: Opinion mining and sentiment analysis. In: Foundatoins and Trends in Information Retrieval, Rome, Italy, pp. 1–135 (September 2008)
9. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), pp. 79–86 (2002)
10. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pp. 339–346 (2005)
11. Snyder, B., Barzilay, R.: Multiple aspect ranking using the good grief algorithm. In: Proceedings of the Joint Conference of the North American Chapter of the Association for Computational Linguistics and Human Language Technologies, pp. 300–307 (2007)
12. Titov, I., McDonald, R.: A joint model of text and aspect ratings for sentiment summarization. In: Proceedings of the 46th Meeting of Association for Computational Linguistics (ACL 2008), pp. 783–792. Morgan Kaufmann, Rome (2008)
13. Turney, P.: Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Meeting of Association for Computational Linguistics (ACL 2002), pp. 417–424 (2002)
14. Wang, H., Lu, Y., Zhai, C.: Latent aspect rating analysis on review text data: A rating regression approach. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD 2010), pp. 783–792 (2010)
15. Zhuang, L., Jing, F., Zhu, X.Y.: Movie review mining and summarization. In: Proceedings of the 15th Conference on Information and Knowledge Management (CIKM 2006), pp. 43–50 (2006)

Clustering-Based *k*-Anonymity

Xianmang He[1], HuaHui Chen[1], Yefang Chen[1],
Yihong Dong[1], Peng Wang[2], and Zhenhua Huang[3]

[1] School of Information Science and Technology, NingBo University
No.818, Fenghua Road, Ning Bo, 315122, P.R. China
{hexianmang,chenhuahui,chenyefang,dongyihong}@nbu.edu.cn
[2] School of Computer Science and Technology, Fudan University
No.220, Handan Road, Shanghai, 200433, P.R. China
pengwang5@fudan.edu.cn
[3] School of Electronic and Information Engineering, TongJi University
NO.1239. Siping Road, Shanghai, 200433, P.R. China
Huangzhenhua@tongji.edu.cn

Abstract. Privacy is one of major concerns when data containing sensitive information needs to be released for ad hoc analysis, which has attracted wide research interest on privacy-preserving data publishing in the past few years. One approach of strategy to anonymize data is generalization. In a typical generalization approach, tuples in a table was first divided into many QI (quasi-identifier)-groups such that the size of each QI-group is no less than k. Clustering is to partition the tuples into many clusters such that the points within a cluster are more similar to each other than points in different clusters. The two methods share a common feature: distribute the tuples into many small groups. Motivated by this observation, we propose a clustering-based k-anonymity algorithm, which achieves k-anonymity through clustering. Extensive experiments on real data sets are also conducted, showing that the utility has been improved by our approach.

Keywords: privacy preservation, algorithm, proximity privacy.

1 Introduction

Privacy leakage is one of major concerns when publishing data for statistical process or data analysis. In general, organizations need to release data that may contain sensitive information for the purposes of facilitating useful data analysis or research. For example, patients' medical records may be released by a hospital to aid the medical study. Records in Table 1 (called the microdata) is an example of patients' records published by hospitals. Note that attribute *Disease* contains sensitive information of patients. Hence, data publishers must ensure that no adversaries can accurately infer the disease of any patient. One straightforward approach to achieve this goal is excluding unique identifier attributes, such as *Name* from the table, which however is not sufficient for protecting privacy leakage under *linking-attack*[1, 2]. For example, the combination of *Age* and

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 405–417, 2012.

Table 1. Microdata T

	Age	Zip	Diesease
Andy	20	25	Flu
Bob	20	30	Bronchitis
Jane	30	25	Gastritis
Alex	40	30	Penumonia
Mary	50	10	Flu
Lily	60	5	Bronchitis
Lucy	60	10	Gastritis

Table 2. Generalization T^*

G-ID	Age	Zip	Diesease
1	[20-20]	[25-30]	Flu
1	[20-20]	[25-30]	Bronchitis
2	[30-40]	[25-30]	Gastritis
2	[30-40]	[25-30]	Penumonia
3	[50-60]	[5-10]	Flu
3	[50-60]	[5-10]	Bronchitis
3	[50-60]	[5-10]	Gastritis

Zipcode can be potentially used to identify an individual in Table 1, and has been called a quasi-identifier (QI for short)[1] in literatures. If an adversary has the background knowledge about Bob, that is: Age=20 and Zipcode=30, then by joining the background knowledge to Table 1, he can accurately infer Bob's disease, that is bronchitis.

To protect privacy against re-identifying individuals by joining multiple public data sources, k-anonymity ($k \geq 2$) was proposed, which requires that each record in a table is indistinguishable from at least $k - 1$ other records with respect to certain quasi-identifiers. Generally, to achieve k-anonymity, generalization [1–3] is a popular methodology of privacy preservation for preventing linking attacks. Enough degree of generalization will hide a record in a crowd with at least k records with the same QI-values, thereby achieving k-anonymity. Table 2 demonstrates a generalized version of Table 1 (e.g., the Zip 30 of Bob, for instance, has been generalized to an interval [25, 30]). The generalization results in 3 equivalence classes, as indicated by their group-IDs. Each equivalence class is referred to as a QI-group. As a result, given Table 2, even if an adversary has the exact QI-values of Bob, s/he still can not exactly figure out the tuple of Bob from the first QI-group.

1.1 Motivation

Although generalization-based algorithms have successfully achieved the privacy protection objective, as another key issue in data anonymization *utility* still needs to be carefully addressed. Great efforts have been dedicated to developing algorithms that improve utility of anonymized data while ensuring enough privacy-preservation. One of the direct measures of the utility of the generalized data is information loss. In order to make the anonymized data as useful as possible for certain applications, it is required to reduce the information loss as much as possible. In general, *the less total information loss leads to better utility, which reflects its usefulness as one of the steps in exploratory data analysis.*

Clustering [4] is a method commonly used to automatically partition a data set into many groups. As an example of clustering is depicted in Figure 1-3. The input points are shown in Figure 1, and the steps to the desired clusters are shown in Figure 2 and Figure 3. Here, points belonging to the same cluster are given the same color.

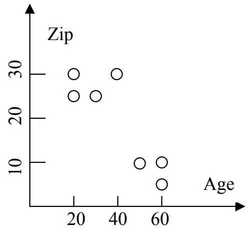


Fig. 1. The points inTable 1

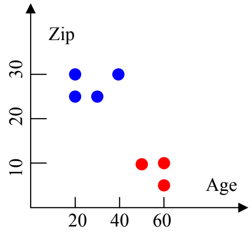


Fig. 2. Data Clustering (Step 1)

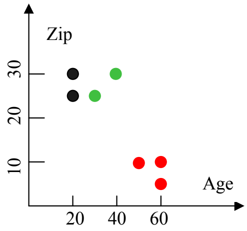


Fig. 3. Data Clustering (Step 2)

Then, we may wonder: *Can we significantly improve the utility while preserving k-anonymity by clustering-based approaches ?* The answer depends on whether it is possible to partition microdata into clusters with less information loss while still ensuring k-anonymity. Intuitively, data points within a cluster are more similar to each other than they are to a point belonging to a different cluster.

The above observation motivates us to devise a new solution to improve the data utility of clustering-based solutions. As an example, we illustrate the details to generalize Table 1 by our approach. Let *gen* be a generalization function that takes as input a set of tuples and returns a generalized domain. Firstly, Table 1 is divided into 2 clusters, denoted by red and blue in Figure 2, respectively. Then, the cluster denoted by blue is further divided into 2 cluster, denoted by black and green color in Figure 3. Finally, tuples with same color are generalized as a QI-group, that is, tuple Andy and Bob consists of the first QI-group, and assign gen({Andy, Bob})=$\langle[20-20],[25-30]\rangle$ to the first QI-group. Similarly, {Jane,Alex}, {Mary, Lily, Lucy} make the second and third QI-group. Eventually, table 2 is the final result by our approach.

In this paper, we mainly focused on the basic k-anonymity model due to the following reasons: (i) k-anonymity is a fundamental model for privacy protection, which has received wide attention in the literatures; (ii) k-anonymity has been employed in many real applications such as location-based services [5, 6], where there are no additional (sensitive) attributes; (iii) There is no algorithm that is suitable for so many privacy metrics such as l-diversity[7], t-Closeness [8], but algorithms for k-anonymity are simple yet effective, and can be further adopted for other privacy metrics. Apart from the k-anonymity model, we also consider the scenarios with stronger adversaries, extending our approach to l-diversity(Section 4)

The rest of the paper is organized as follows. In Section 2, we give the definitions of basic concept and the problem will be addressed in this paper. In Section 3, we present the details of our generalization algorithm. Section 4 discusses the extension of our methodology for l-diversity. We review the previously related research in Section 5. In Section 6, we experimentally evaluate the efficiency and effectiveness of our techniques. Finally, the paper is concluded in Section 7.

2 Fundamental Definitions

Let T be a microdata table that contains the private information of a set of individuals and has d QI-attributes $A_1, ..., A_d$, and a sensitive attribute A_s. We consider that A_s is numerical, and every QI-attribute $A_i(1 \leq i \leq d)$ can be either numerical or categorical. All attributes have finite and positive domains. For each tuple $t \in T, t.A_i(1 \leq i \leq d)$ denotes its value on A_i, and $t.A_s$ represents its SA value.

2.1 Basic Concept

A *quasi-identifier* $QI = \{A_1, A_2, \cdots, A_d\} \subseteq \{A_1, A_2, \cdots, A_n\}$ is a minimal set of attributes, which can be joined with external information in order to reveal the personal identity of individual records.

A *partition* P consists of several subsets $G_i(1 \leq i \leq m)$ of T, such that each tuple in T belongs to exactly one subset and $T = \bigcup_i^m G_i$. We refer to each subset G_i as a QI-group.

2.2 *K*-means Clustering

K-means clustering [4] is a method commonly used to automatically partition a data set into K groups. It proceeds by selecting K initial cluster centers and then iteratively refining them as follows:

Step 1. Each tuple t_i is assigned to its closest cluster center.

Step 2. Each cluster center C_j is updated to be the mean of its constituent instances.

The algorithm converges when there is no further change in assignment of tuples to clusters. In this work, we initialize the clusters using instances picked at random from the data set. The data sets we used are composed solely of either numeric features or categorical features. For both numeric and categorical features, we adopt the normalized certainty penalty(see the definition 1) to measure the distance.

The final issue is how to choose K. To keep the algorithm simple in this paper, we consider binary partitioning, that is, K is fixed as 2.

2.3 Problem Definition

Some methods have been developed to measure the information loss in anonymization. In this paper, we adopt the normalized certainty penalty to measure the information loss.

Definition 1 (Normalized Certainty Penalty [9]). *Suppose a table T is anonymized to T^*. In the domain of each attribute in T, suppose there exists a* *global order on all possible values in the domain. If a tuple t in T^* has range $[x_i, y_i]$ on attribute $A_i (1 \leq i \leq d)$, then the normalized certainty penalty in t on A_i is $NCP_{A_i}(t) = \frac{|y_i - x_i|}{|A_i|}$, where $|A_i|$ is the domain of the attribute A_i. For tuple t, the normalized certainty penalty in t is $NCP(t) = \sum_i^d w_i \cdot NCP_{A_i}(t)$, where w_i is the weight of attribute A_i. The normalized certainty penalty in T is $\sum_{t \in T^*} NCP(t)$.*

Now, we are ready to give the formal definition about the problem that will be addressed in this paper. Information loss is an unfortunate consequence of data anonymization. We aim to generate a utility-friendly version anonymizaiton for a microdata such that the privacy can be guaranteed by k-anonymity and the information loss quantified by NCP is minimized. Now, we are ready to give the formal definition about the problem that will be addressed in this paper. (Limited by space, all proofs are omitted.)

Definition 2 (Problem Definition). *Given a table T and an integer k, anonymize it by clustering to be T^* such that T^* is k-anonymity and the total information loss is minimized measured by NCP.*

Theorem 1. *(Complexity) The problem of optimal clustering-based anonymization is NP-hard under the metric NCP.*

3 Clustering-Based Generalization Algorithm

In this section, we will present the details of our clustering-based anonymization approach. The key of our algorithm is to divide all tuples into more compact clusters efficiently and correctly. We now proceed to a discussion of our modifications to the K-means algorithm.

To keep the algorithm simple, we consider binary clustering. That is, in each round, we partition a set of tuples into two subsets by clustering. In order to reduce the total information loss, we will cluster the microdata following the idea: *distribute tuples sharing the same or quite similar QI-attributes into the same cluster*. We adopt the NCP to measure the distance. The detailed partitioning procedure is presented in Figure 4. Initially, S contains T itself (line 1); then, each $G \in S$ is divided into two generalizable subsets G_1 and G_2 such that $G_1 \cup G_2 = G, \ G_1 \cap G_2 = \emptyset$ (line 5-7).

The size of the two subsets should $\geq k$, otherwise adjustment is needed (line 8). Without loss of generality, assume that $G_1 < k$, we need to borrow $k - |G_1|$ tuples from G_2 to make sure that G_1 has a cardinality $\geq k$.

The tries to converges will cost unacceptable time, to accelerate the partitioning, the attempts to cluster G are tried r times and tuples of G are randomly shuffled for each time (line 4). Our experimental results show that most of G can be partitioned into two sub-tables by up to $k = 15$ tries. The algorithm stops when no sub-tables in S can be further partitioned.

By the Lemma in the paper [9, 10] that the optimal k-anonymity partitioning of microdata does not contain groups of more than $2k - 1$ records, we have that the partitioning algorithm will terminate when the size of all groups is between k and $2k-1$. If at least one group contains a cardinality more than $2k-1$, the partitioning algorithm will continue.

In the above procedure, the way that we partition G into two subsets G_1 and G_2 is influential on the information loss of the resulting solution. In the first round, we randomly choose two tuples t_1, t_2 as the center points C_1, C_2 , and then insert them G_1 and G_2 separately. Then, we distribute each tuple $w \in G$: for each tuple w, we compute $\varDelta_1 = NCP(C_1 \cup w)$ and $\varDelta_2 = NCP(G_2 \cup w)$, and add tuple w to the group that leads to lower penalty (line 7). If $\varDelta_1 = \varDelta_2$, assign the tuple to the group who has lower cardinality. After successfully partitioning G, remove the tuples t_1 and t_2 from $G_1 - \{t_1\}$ and $G_2 - \{t_2\}$. At the later each round, the center points C_i are conducted as follows: $C_i = \frac{\sum_{t \in G_i} t}{|G_i|}, i = 1, 2.$ that is, for each attribute $A_j (1 \leq j \leq d)$, $C_i.A_j = \frac{\sum_{t \in G_i} t.A_j}{|G_i|}, i = 1, 2.$

After the each partition, if the current partition is better than previous tries, record the partition result G_1, G_2 and the total sum of $NCP(G_1)$ and $NCP(G_2)$. That is, we pick the one that that minimizes the sum of $NCP(G_1)$ and $NCP(G_2)$ as the final partition among the r partitions(line 9). Each round of G can be accomplished in O$(r \cdot (|G| \cdot (6 + \lambda)))$ expected time, where λ is the cost of evaluating loss. The computation cost is theoretically bounded in Theorem 2.

Theorem 2. *For microdata T, the clustering-based algorithm can be accomplished in $O(r \cdot |T| \cdot log(|T|))$ average time, where r is the number of rounds , and $|T|$ is the cardinality of microdata T.*

```
Input: A microdata $T$, integers $k$ and rounds $r$
Output: anonymized table $T^*$;
Method:
/* the parameter $r$ is number of rounds to cluster $G$*/
1. $S = \{T\}$;
2. While($\exists G \in S$ that $|G| \geq 2k$)
3.     For $i = 1$ to $r$
4.         Randomly shuffle the tuples of $G$;
5.         Set Center $C_i = \frac{\sum_{t \in G_i} t}{|G_i|}$;
6.         Set $G_1 = G_2 = \emptyset$;
7.         Distribute each tuple $w$ in $G$:
               compute $\varDelta_1 = NCP(w \cup C_1)$ and $\varDelta_2 = NCP(w \cup C_2)$;
               If($\varDelta_1 < \varDelta_2$) then add $w$ to $G_1$, else add $w$ to $G_2$;
8.         Adjust $G_1$, $G_2$ that each group has at least $k$ tuples;
9.         If the current partition is better than previous tries, record $G_1$ and $G_2$;
10.    Remove $G$ from $S$, and add $G_1, G_2$ to $S$;
11. Return $S$;
```

Fig. 4. The partitioning algorithm

4 Extension to l-Diversity

In this section, we discuss how we can apply clustering-based anonymization for other privacy principles. In particular, we focus on l-diversity, described in Definition 3.

Definition 3 (l-diversity[7]). *A generalized table T^* is l-diversity if each QI-group $QI_i \in T^*$ satisfies the following condition: let v be the most frequent A_s value in QI_i, and $c_i(v)$ be the number of tuples $t \in QI_i$, then $\frac{c_i(v)}{|QI_i|} \leq \frac{1}{l}$.*

To generalize a table through clustering-based anonymization, we partition a table into sub-tables T_i which satisfy l-diversity: after each round of the above partitioning, if both (G_1 and G_2)) satisfy l-diversity, we remove G from S, and add G_1, G_2 to S; otherwise G is retained in S. Then for each subset $T_i \in S$, we conduct the splitting algorithm (see Figure 5) to produce the final l-diverse partitions.

The principle l-diversity demands that: the number of the most frequent A_s value in each QI-group QI_i can't exceed $\frac{|QI_i|}{l}$. Motivated by this, we arrange the tuples to a list ordered by its A_s values, then distribute the tuples in L into $QI_i (1 \leq i \leq g)$ a round-robin fashion. The resulting splitting is guaranteed to be l-diversity, which is stated in Theorem 3. (If table T with sensitive attribute A_s satisfies $\max\{c(v) : v \in T.A_s\} > \frac{|T|}{l}$, then there exists no partition that is l-diversity.)

```
Input: table $T$, parameter $l$
Output: QI-groups $QI_j$ that satisfy $l$-diversity;
Method:
1. If $\max\{c(v) : v \in T.A_s\} \geq \frac{|T|}{l}$, Return;
2. Hash the tuples in $T$ into groups $Q_1, Q_2, \cdots, Q_\lambda$ by their $A_s$ values;
3. Insert these groups $Q_1, Q_2, \cdots, Q_\lambda$ into a list $L$ in order;
4. Let $g = \frac{|T|}{l}$, set QI-groups $QI_1 = QI_2 = \cdots = QI_g = \emptyset$;
5. Assign tuple $t_i \in L$ $(1 \leq i \leq |L|)$ to $QI_j$, where $j = (i \bmod g) + 1$
```

Fig. 5. The splitting algorithm

Theorem 3. *If table T with sensitive attribute A_s satisfies* $\max\{c(v) : v \in T.A_s\} \leq \frac{|T|}{l}$ *(where $c(v)$ is the number of tuples in T with sensitive value v), the partition produced by our splitting algorithm fulfills l-diversity.*

5 Related Work

In this section, previous related work will be surveyed. Existing generalization algorithms can be further divided into heuristic-based and theoretical-based approaches. Generally, appropriate heuristics are general so that they can be used

Table 3. Summary of attributes

Attribute	Number of distinct values	Types
Age	78	Numerical
Gender	2	Categorical
Education	17	Numerical
Marital	6	Categorical
Race	9	Numerical
Work-class	10	Categorical
Country	83	Numerical
Occupation	50	Sensitive

(a) SAL

Attribute	Number of distinct values	Types
Age	78	Numerical
Occupation	711	Numerical
Birthplace	983	Numerical
Gender	2	Categorical
Education	17	Categorical
Race	9	Categorical
Work-class	9	Categorical
Marital	6	Categorical
Income	[1k,10k]	Sensitive

(b) INCOME

Table 4. Parameters and tested values

Parameter	Values
k	250,200,150,**100**,50
cardinality *n*	**100k**,200k,300k,400k,500k
number of QI-attributes *d*	3,4,5,**6**

in many anonymization models. To reduce information loss, efficient greedy solutions following certain heuristics have been proposed [9–13] to obtain a near optimal solution. Generally, these heuristics are general enough to be used in many anonymization models. Incognito [14] provides a practical framework for implementing full-domain generalization, borrowing ideas from frequent item set mining, while [10] presents a framework mapping the multi-dimensional quasi-identifiers to 1-Dimensional(1-D) space. For 1-D quasi-identifiers, an algorithm of $O(K \cdot N)$ time complexity for optimal solution is also developed. It is discovered that k-anonymizing a data set is strikingly similar to building a spatial index over the data set, so that classical spatial indexing techniques can be used for anonymization [15]. To achieve k-anonymity, Mondrian [16] takes a partitioning approach reminiscent of KD-trees.

The idea of non-homogeneous generalization was first introduced in [17], which studies techniques with a guarantee that an adversary cannot associate a generalized tuple to less than K individuals, but suffering additional types of attack. Authors of paper [13] proposed a randomization method that prevents such type of attack and showed that k-anonymity is not compromised by it, but its partitioning algorithm is only a special of the top-down algorithm presented in [9]. The model of the paper [13, 17], the size of QI-groups is fixed as 1.

The algorithms mentioned above work well on practical data sets, but do not have attractive asymptotical performance in the worst case. This motivates studies on the theoretical aspects of k-anonymity [16, 18]. Most of these works show that the problem of optimal k-anonymity is NP-hard even a simple quality metric is employed.

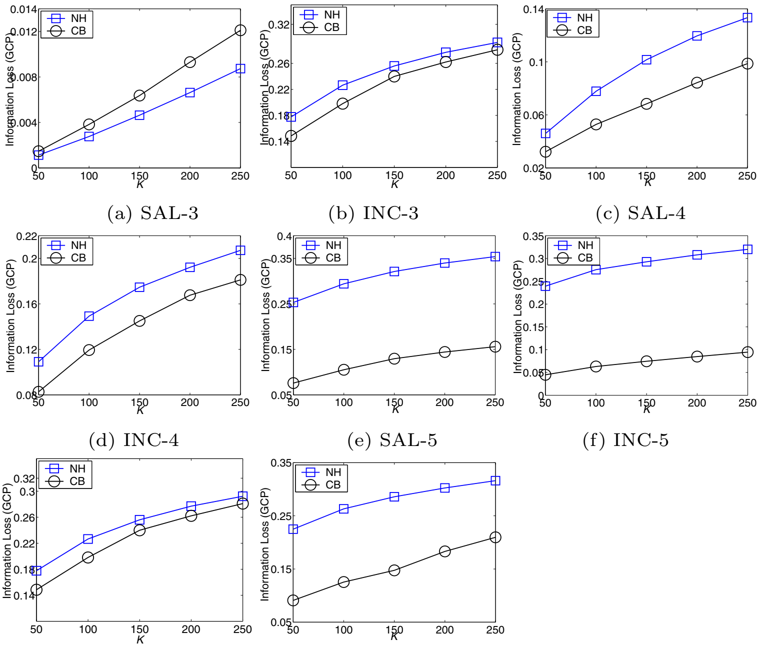
6 Empirical Evaluation

In this section, we will experimentally evaluate the effectiveness and efficiency of the proposed techniques. Specifically, we will show that by our technique (presented in Section 3) have significantly improved the utility of the anonymized data with quite small computation cost.

Towards this purpose, two widely-used real databases sets: SAL and INCOME(downloadable from http://ipums.org) with 500k and 600k tuples, respectively, will be used in following experiments. Each tuple describes the personal information of an American. The two data sets are summarized in Table 3.

In the following experiments, we compare our cluster-based anonymity algorithm (denoted by CB) with the existing state-of-the-art technique: the non-homogeneous generalization [13](NH for short). (The fast algorithm [10] was cited and compared with NH in the paper [13], therefore, we omit the details of the fast algorithm.)

In order to explore the influence of dimensionality, we create two sets of microdata tables from SAL and INCOME. The first set has 4 tables, denoted as SAL-3, $\cdots$, SAL-6, respectively. Each SAL-d ($3 \leq d \leq 6$) has the first d attributes in Table 3 as its QI-attributes and Occupation as its sensitive attribute(SA). For example, SAL-4 is 5-Dimensional, and contains QI-attributes: Age, Gender, and



(a) SAL-3

(b) INC-3

(c) SAL-4

(d) INC-4

(e) SAL-5

(f) INC-5

(g) SAL-6

(h) INC-6

Fig. 6. The Global Certainty Penalty vs. parameters k and d

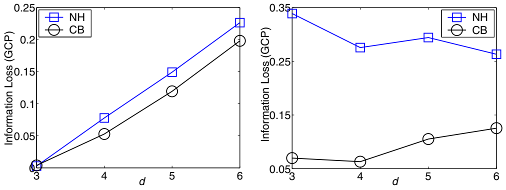
Education, Marital. The second set also has 4 tables INC-3, $\cdots$, INC-6, where each INC-d ($3 \leq d \leq 6$) has the first d attributes as QI-attributes and income as the SA.

In the experiments, we investigate the influence of the following parameters on information loss of our approach: (i) value of k in k-anonymity; (ii)number of attributes d in the QI-attributes; (iii)number of tuples n. Table 4 summarizes the parameters of our experiments, as well as their values examined. **Default values are in bold font**. Data sets with different cardinalities n are also generated by randomly sampling n tuples from the *full* SAL-d or INC-d ($3 \leq d \leq 6$). All experiments are conducted on a PC with 1.9 GHz AMD Dual Core CPU and 1 gigabytes memory. All the algorithms are implemented with VC++ 2008.

We measure the information loss of the generalized tables using GCP, which is first used in [10]. Note that GCP essentially is equivalent to NCP with only a difference of constant number $d \times N$. Specifically, under the same partition P of table T, $GCP(T) = \frac{NCP(T)}{d \times N}$(d is the size of QI-attributes), when all the weights are set to 1.0.

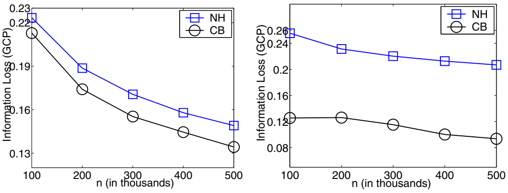
6.1 Privacy Level *K*

In order to study the influence of k on data utility, we observe the evolution of GCP that has been widely used to measure the information loss of the generalized tables by varying k from 50 to 250 with the increment of 50. In all following experiments, without explicit statements, default values in Table 4 will be used for all other parameters. The results on SAL-d and INC-d ($3 \leq d \leq 6$) data are



(a) SAL-d (b) INC-d

Fig. 7. The Global Certainty Penalty vs. QI-Dimensionality d



(a) SAL-7 (b) INC-7

Fig. 8. The Global Certainty Penalty vs. Cardinality n

shown in Figure 6 (a)-6(h). From the results, we can clearly see that information loss of CB sustains a big improvement over NH, for the tested data except the on SAL-3. Another advantage of our model over NH is that the utility achieved by our model is less sensitive to domain size than NH. From the figures, we can see that data sets generated by NH has a lower GCP on SAL-d than that on INC-d ($4 \leq d \leq 7$) due to the fact that domain size of SAL is smaller than that of INC. Such a fact implies that the information loss of NH is positively correlated to the domain size. However, in our model, domain size of different data set has less influence on the information loss of the anonymized data.

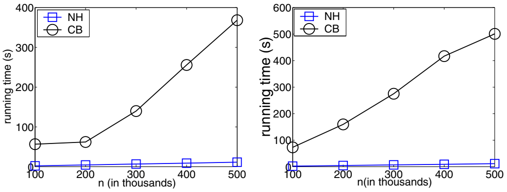
Results of this experiment also suggest that for almost all tested data sets the GCP of these algorithms grows linearly with k. This can be reasonably explained since larger k will lead to more generalized QI-groups, which inevitably will sacrifice data utility. NH performs well when the dimensionality of QI-Attributes is low and the domain size is small, see the experiment results in the paper[13].

6.2 QI-Attributes Dimensionality d

Experiments of this subsection is designed to show the relation between the information loss of these algorithms and data dimensions d. In general, the information loss will increase with d, since data sparsity or more specifically the data space characterized by a set of attributes exponentially increases with the number of attributes in the set, i,e, dimensions of the table. Figure 7(a) and 7(b) compare the information loss of the anonymization generated by the these four methods with respect to different values of d on SAL-d and INC-d, respectively. It is clear that the anonymization generated by the cluster-based method has a lower global certainty penalty compared to that of NH. The advantage of CB is obvious, and such an advantage of CB can be consistently achieved when d lies between 4 to 6.

6.3 Cardinality of Data Set n

In this subsection, we investigate the influence of the the table size n on information loss. The results of experiments on two data sets SAL-7 and INC-7 are shown in Figure 8(a) and 8(b), respectively. We can see that the information



(a) SAL-7 (b) INC-7

Fig. 9. Running time vs. Cardinality n

loss of these methods on both two data sets decreases with the growth of n. This observation can be attributed to the fact that when the table size increases more tuples will share the same or quite similar QI-attributes. As a result, it is easier for the partitioning strategies to find very similar tuples to generalize. Similar to previously experimental results, our method is the clear winner since information loss of CB is significantly small than that of NH, which is consistently observed for various database size.

6.4 Efficiency

Finally, we evaluate the overhead of performing anonymization. Figure 9(a) and 9(b) show the computation cost of the these anonymization methods on two data sets, respectively. We compare CB with NH when evaluating computational cost. The running time of tow algorithms increases linearly when n grows from 100k to 500k, which is expected since more tuples that need to be anonymized will cost longer time to finish the anonymization procedure. The NH method is more efficient. Comparison results show that the advantages of our method in anonymization quality do not come for free. However, in the worst case, our algorithm can be finished in 500 seconds, which is acceptable. In most real applications quality is more important than running time, which justifies the strategy to sacrifice certain degree of time performance to achieve higher data utility.

Summary. Above results clearly show that clustering-based anonymization achieves less information loss than the non-homogeneous anonymization (NH) in cases where the dimensionality of QI-attribute $d > 3$. NH has a good performance when the domain size is small, and the dimensionality of QI-Attributes is low. This is due to its greedy partitioning algorithm.

7 Conclusion

As privacy becomes a more and more serious concern in applications involving microdata, good anonymization is of significance. In this paper, we propose an algorithm which is based on clustering to produce a utility-friendly anonymized version of microdata. Our extensive performance study shows that our methods outperform the non-homogeneous technique where the size of QI-attribute is larger than 3.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China (NO.6090303, NO.60902097, NO.60973047), the Natural Science Foundation of Zhejiang Province of China under Grant No. Y1091189 and No.Y1090571, the Research Fund for the Doctoral Program of Higher Education(No. 20090072120056), the Special Fund for Information Development of Shanghai (No. 200901015), the National High-Tech Research and Development Plan of China (No. 2008AA04Z106), the Project of Science and Technology Commission of Shanghai Municipality (NO.08DZ1122300), and the Major Scientific & Technology Specific Programs of Zhejiang Province for Key Industrial Project(No.2011C11042).

References

1. Sweeney, L.: k-anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10(5), 557–570 (2002)
2. Samarati, P.: Protecting respondents' identities in microdata release. TKDE 13(6), 1010–1027 (2001)
3. Samarati, P., Sweeney, L.: Generalizing data to provide anonymity when disclosing information. In: PODS 1998, p. 188. ACM, New York (1998)
4. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations, Berkeley, pp. 281–297 (1967)
5. Kalnis, P., Ghinita, G., Mouratidis, K., Papadias, D.: Preventing location-based identity inference in anonymous spatial queries. TKDE 19(12), 1719–1733 (2007)
6. Mokbel, M.F., Chow, C.-Y., Aref, W.G.: The new casper: query processing for location services without compromising privacy. In: VLDB 2006, pp. 763–774 (2006)
7. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: l-diversity: Privacy beyond k-anonymity. In: ICDE 2006, p. 24 (2006)
8. Li, N., Li, T.: t-closeness: Privacy beyond k-anonymity and l-diversity. In: KDD 2007, pp. 106–115 (2007)
9. Xu, J., Wang, W., Pei, J., Wang, X., Shi, B., Fu, A.W.-C.: Utility-based anonymization using local recoding. In: KDD 2006, pp. 785–790. ACM (2006)
10. Ghinita, G., Karras, P., Kalnis, P., Mamoulis, N.: Fast data anonymization with low information loss. In: VLDB 2007, pp. 758–769. VLDB Endowment (2007)
11. Fung, B.C.M., Wang, K., Yu, P.S.: Top-down specialization for information and privacy preservation, pp. 205–216 (2005)
12. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Workload-aware anonymization. In: KDD 2006, pp. 277–286. ACM, New York (2006)
13. Wong, W.K., Mamoulis, N., Cheung, D.W.L.: Non-homogeneous generalization in privacy preserving data publishing. In: SIGMOD 2010, pp. 747–758. ACM, New York (2010)
14. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Incognito: efficient full-domain k-anonymity. In: SIGMOD 2005, pp. 49–60. ACM, New York (2005)
15. Iwuchukwu, T., Naughton, J.F.: K-anonymization as spatial indexing: toward scalable and incremental anonymization. In: VLDB 2007, pp. 746–757 (2007)
16. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Mondrian multidimensional k-anonymity. In: ICDE 2006, Washington, DC, USA, p. 25 (2006)
17. Gionis, A., Mazza, A., Tassa, T.: k-anonymization revisited. In: ICDE 2008, pp. 744–753. IEEE Computer Society, Washington, DC (2008)
18. Bayardo, R.J., Agrawal, R.: Data privacy through optimal k-anonymization. In: ICDE 2005, pp. 217–228. IEEE Computer Society, Washington, DC (2005)

Unsupervised Ensemble Learning for Mining Top-n Outliers[⋆]

Jun Gao[1], Weiming Hu[1], Zhongfei(Mark) Zhang[2], and Ou Wu[1]

[1] National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing, China
{jgao,wmhu,wuou}@nlpr.ia.ac.cn

[2] Dept. of Computer Science, State Univ. of New York at Binghamton,
Binghamton, NY 13902, USA
zhongfei@cs.binghamton.edu

Abstract. Outlier detection is an important and attractive problem in knowledge discovery in large datasets. Instead of detecting an object as an outlier, we study detecting the n most outstanding outliers, i.e. the top-n outlier detection. Further, we consider the problem of combining the top-n outlier lists from various individual detection methods. A general framework of ensemble learning in the top-n outlier detection is proposed based on the rank aggregation techniques. A score-based aggregation approach with the normalization method of outlier scores and an order-based aggregation approach based on the distance-based Mallows model are proposed to accommodate various scales and characteristics of outlier scores from different detection methods. Extensive experiments on several real datasets demonstrate that the proposed approaches always deliver a stable and effective performance independent of different datasets in a good scalability in comparison with the state-of-the-art literature.

1 Introduction

Outlier detection is an important knowledge discovery problem in finding unusual events and exceptional cases from large datasets in many applications such as stock market analysis, intrusion detection, and medical diagnostics. Over the past several decades, the research on outlier detection varies from the global computation to the local analysis, and the descriptions of outliers vary from binary interpretations to probabilistic representations. Global outlier detection [3,4,5] identifies an observational object with a binary label by the global computation. Local outlier detection [6,7,8,9] provides a probabilistic likelihood called outlier score to capture how likely an object is considered as an outlier. Outlier scores can be used not only to discriminate outliers from normal data, but also to rank all the data in a database, such as the top-n outlier detection. There are other efforts that transform the unsupervised outlier detection to a classification via artificially generated outliers [10].

[⋆] This work is supported in part by NSFC(Grant No. 60825204, 60935002 and 60903147), NBRPC(2012CB316400) and US NSF(IIS-0812114, CCF-1017828) as well as the Zhejiang University – Alibaba Financial Joint Lab.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 418–430, 2012.

Although there are numerous outlier detection methods proposed in the literature, no one method performs better than the others under all circumstances, and the best method for a particular dataset may not be known a priori. Each detection method is proposed based on the specific priori knowledge. For example, the nearest neighbor based methods assume that the feature space is well enough to discriminate outliers from normal data, while the classification based and the statistical methods need to suppose the distributions of outliers and normal objects, respectively. Hence, their detection performances vary with the nature of data. This setting motivates a fundamental information retrieval problem - the necessity of an ensemble learning of different detection methods to overcome their drawbacks and to increase the generalization ability, which is similar to *meta-search* that aggregates query results from different search engines into a more accurate ranking. Like *meta-search*, ensemble learning in the top-n outlier detection is more valuable than the fusion of the binary labels, especially in large databases. There is the literature on the ensemble learning of outlier detection, such as [13,14,15]. However, all these efforts state the problem of effectively detecting outliers in the sub-feature spaces. Since the work of Lazarevic and others focuses on the fusion of the sub-feature spaces, these methods are very demanding in requiring the full spectrum of outlier scores in the datasets that prevents them from the fusion of the top-n outlier lists in many real-world applications.

Although the problem of ensemble learning in the top-n outlier detection shares a certain similarity to that of *meta-search*, they have two fundamental differences. First, the top-n outlier lists from various individual detection methods include the order information and outlier scores of n most outstanding objects. Different detection methods generate outlier scores in different scales. This requires the ensemble framework to provide a unified definition of outlier scores to accommodate the heterogeneity of different methods. Second, the order-based rank aggregation methods, such as Mallows Model [18], can only combine the information of the order lists with the same length, which prevents the application of these rank aggregation methods in the fusion of top-k outlier lists. Because, for a particular dataset, there are always several top-k outlier lists with various length used to measure the performance and effectiveness of a basic outlier detection method. In order to address these issues, we propose a general framework of ensemble learning in the top-n outlier detection shown in Figure 1, and develop two fusion methods: the score-based aggregation method (*SAG*) and the order-based aggregation method (*OAG*). To the best of our knowledge, this is the first attempt to the ensemble learning in the top-n outlier detection. Specifically, the contributions of this paper are as follows:

- We propose a score-based aggregation method (*SAG*) to combine the top-n outlier lists given by different detection methods without supervision. Besides, we propose a novel method for transforming outlier scores to posterior probabilities, which is used to normalize the heterogeneous outlier scores.
- We propose an order-based aggregation method (*OAG*) based on the distanced-based Mallows model [16] to aggregate the different top-n outlier lists without supervision, which can deal with the fusion of top-k outlier lists with various length. This method only adopts the order information, which avoids the normalization of outlier scores.

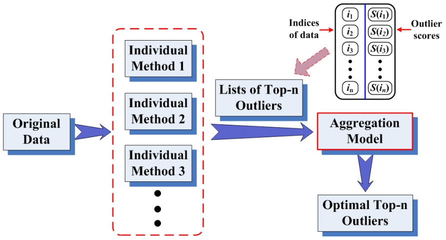


Fig. 1. The general framework of ensemble learning

- Extensive experiments on real datasets validate the effectiveness of these aggregation methods, where several state-of-the-art outlier detection methods, including the nearest neighbor based and the classification based methods, are selected as the individual methods for the ensemble learning. Besides, the robustness of the proposed aggregation methods is evaluated based on the Uniform noise and the Gaussian noise.

The remainder of this paper is organized as follows. Section 2 introduces the framework of ensemble learning in the top-n outlier detection and the two novel aggregation methods: the score-based and the order-based methods. Section 3 reports the experimental results. Finally, Section 4 concludes the paper.

2 Methodologies

We first introduce the general framework and the basic notions of ensemble learning in the top-n outlier detection, and then introduce the score-based method with a unified outlier score and the order-based method based on the distance-based Mallows model, respectively.

2.1 Framework and Notions of Ensemble Learning

Let $X = [x_1, x_2, x_3, \ldots, x_d]$ be an object in a dataset D, where d is the number of attributes and $|D|$ is the number of all the objects.

As shown in Figure 1, there are K individual detection methods that process the original data in parallel. Essentially, all the individual methods return outlier scores rather than binary labels to generate the top-n outlier lists, where the number n is determined by users. The top-n outlier list σ_i assigned to the i-th individual method is represented as $(\sigma^{-1}(1), S(i_1); \cdots; \sigma^{-1}(n), S(i_n))$, where $\sigma^{-1}(i)$ denotes the index of the object assigned to rank i and $S(\sigma^{-1}(i))$ is its outlier score. Correspondingly, $\sigma(i)$ is the rank assigned to object X_i. Let R_n be the set of all the top-n orderings over $|D|$ objects, and $d : R_n \times R_n \longrightarrow \mathbb{R}$ be the distance between two top-n lists, which should be a right-invariant metric. This means that the value of $d(\pi, \sigma)|\forall \pi, \sigma \in R_n$ does not depend on how objects are indexed.

The aggregation model combines K orderings $\{\sigma_i\}_{i=1}^{K}$ to obtain the optimal top-n outlier list. Clearly, the literature with respect to the fusion of sub-feature spaces [13,14,15] can be included in this framework by using the detection model in a special sub-feature space as an individual method. In this paper, we only focus on the unsupervised aggregation models based on the order information and outlier scores.

2.2 Score-Based Aggregation Approach (SAG)

Since a top-n outlier list σ_i contains the order information and the corresponding outlier scores, it is straightforward that combining these outlier scores from different methods improves the detection performance. As mentioned in the previous section, outlier scores of the existing methods have different scales. For example, outlier scores vary from zero to infinity for the nearest based method [6], while lying in the interval $[-1, 1]$ for the classification based method [10]. In this subsection, an effective method is proposed to transform outlier scores to posterior probability estimates. Compared with outlier scores, the posterior probability based on Bayes' theorem provides a robust estimate to the information fusion and a spontaneous measure of the uncertainty in outlier prediction. Without loss of generality, we assume that the higher $S(i)$, the more probable X_i to be considered as an outlier. Let Y_i be the label of X_i, where $Y_i = 1$ indicates that X_i is an outlier and $Y_i = 0$ if X_i is normal. According to Bayes' theorem,

$$P(Y_i = 1|S(i)) = \frac{P(S(i)|Y_i = 1)P(Y_i = 1)}{\sum_{l=0}^{1} P(S(i)|Y_i = l)P(Y_i = l)} = \frac{1}{1 + \frac{P(S(i)|Y_i=0)P(Y_i=0)}{P(S(i)|Y_i=1)P(Y_i=1)}} \quad (1)$$

Let $\varphi(i) = \frac{P(S(i)|Y_i=0)P(Y_i=0)}{P(S(i)|Y_i=1)P(Y_i=1)}$. $ln\big(\varphi(i)\big)$ can be considered as the discriminant function that classifies X_i as normal or outlier. Hence, $ln\big(\varphi(i)\big)$ can be simplified to a linear function, proportional to the Z-Score of $S(i)$ as follows:

$$\varphi(i) = exp\bigg(- \frac{S(i) - \mu}{std} + \tau \bigg) \quad (2)$$

where μ and std are the mean value and standard deviation of the original outlier scores, respectively. In large datasets, these statistics can be computed by sampling the original data. As a discriminant function, $ln\big(\varphi(i)\big) < 0$ means $(S(i) - \mu)/std > \tau$; the object X_i can be assigned as an outlier. In all the experiments, the default value of τ equals 1.5 based on Lemma 1.

Lemma 1: *For any distribution of outlier score $S(i)$, it holds that*

$$P\bigg(\frac{S(i) - \mu}{std} > \tau\bigg) \leq \frac{1}{\tau^2}$$

Proof: *According to Chebyshev's inequality, it holds that,*

$$P\bigg(\frac{S(i) - \mu}{std} > \tau\bigg) \leq P\bigg(|S(i) - \mu| > \tau \cdot std\bigg) \leq \frac{std^2}{(\tau \cdot std)^2} = \frac{1}{\tau^2}$$

Lemma 1 shows a loose bound of deviation probability regardless of the distribution of outlier scores. Supposing that outlier scores follow a normal distribution, $\tau = 1.5$ means that much less than 10% of the objects deviate from the majority of data, which follows the definition of Hawkins outlier [1].

For a top-n outlier list σ_i, objects in the dataset may not be ranked by σ_i. The simple average posterior probabilities are not appropriate to the top-n ranking aggregation. Clearly, objects that appear in all the ranking lists should be more probable to be outliers than ones that are only ranked by a single list. Hence, we apply the following fusion rules which are proposed by Fox and Show [12].

$$rel(i) = n_d^r \sum_j rel_j(i) \quad r \in (-1, 0, 1) \tag{3}$$

where n_d is the number of the orderings that contain object X_i and $rel_j(i)$ is the normalized outlier score of X_i by the j-th individual method. When $r = 1$, the ultimate outlier score is composed of the number of the orderings n_d and the sum of its outlier scores. When $r = 0$, the result is only the sum of its outlier scores. When $r = -1$, it is equivalent to the average outlier scores of the orderings containing X_i. According to Eq. 1 and Eq. 2, the posterior probabilities can be used to normalize outlier scores directly. The detailed steps of *SAG* are shown in Algorithm 1.

Algorithm 1. Score-based aggregation method (*SAG*)

```
Input: $\psi = \{\sigma_k\}_{k=1}^{K}, \gamma$
1. Transform outlier scores in $\psi$ to posterior probabilities according to Eq. {1,2}.
2. Construct an union item pool $U$ including all objects in $\psi$, and denote the size of $U$ as $|U|$.
3. Compute the normalized outlier score $\{rel(i)\}_{i=1}^{|U|}$ for each object in $U$ according to Eq. 3.
4. Sort objects in $U$ based on the normalized outlier scores, and output the optimal list $\pi$.
Output: $\pi$
```

2.3 Order-Based Aggregation Approach (OAG)

Given a judge ordering σ and its expertise indicator parameter θ, the Mallows model [16]generates an ordering π given by the judge according to the formula:

$$P(\pi|\theta, \sigma) = \frac{1}{Z(\sigma, \theta)} exp(\theta \cdot d(\pi, \sigma)) \tag{4}$$

where

$$Z(\sigma, \theta) = \sum_{\pi \in R_n} exp(\theta \cdot d(\pi, \sigma)) \tag{5}$$

According to the right invariance of the distance function, the normalizing constant $Z(\sigma, \theta)$ is independent of σ, which means $Z(\sigma, \theta) = Z(\theta)$. The parameter θ is a non-positive quantity and the smaller the value of θ, the more concentrated at σ the ordering π. When θ equals 0, the distribution is uniform meaning that the ordering given by the judge is independent of the truth.

An extended Mallows model is proposed in [17] as follows:

$$P(\pi|\boldsymbol{\theta},\boldsymbol{\sigma}) = \frac{1}{Z(\boldsymbol{\sigma},\boldsymbol{\theta})}P(\pi)exp\Big(\sum_{i=1}^{K}\theta_i \cdot d(\pi,\sigma_i)\Big) \tag{6}$$

where $\boldsymbol{\sigma} = (\sigma_1,\cdots,\sigma_K) \in R_n^K$, $\boldsymbol{\theta} = (\theta_1,\cdots,\theta_K) \in \mathbb{R}^K$, $P(\pi)$ is a prior, and the normalizing constant

$$Z(\boldsymbol{\sigma},\boldsymbol{\theta}) = Z(\boldsymbol{\theta}) = \sum_{\pi\in R_n} P(\pi)exp\Big(\sum_{i=1}^{K}\theta_i \cdot d(\pi,\sigma_i)\Big) \tag{7}$$

In this extended model, each ordering σ_i is returned by a judge for a particular set of objects. θ_i represents the expertise degree of the i-th judge. Eq. 6 computes the probability that the true ordering is π, given the orderings $\boldsymbol{\sigma}$ from K judges and the degrees of their expertise.

Based on the hypothesis of the distance-based Mallow model, we propose a generative model of *OAG*, which can be described as follows:

$$P(\pi,\boldsymbol{\sigma}|\boldsymbol{\theta}) = P(\boldsymbol{\sigma}|\boldsymbol{\theta},\pi)P(\pi|\boldsymbol{\theta}) = P(\pi)\prod_{i=1}^{K}P(\sigma_i|\theta_i,\pi) \tag{8}$$

The true list π is sampled from the prior distribution $P(\pi)$ and σ_i is drawn from the Mallows model $P(\sigma_i|\theta_i,\pi)$ independently. For the ensemble learning of top-n outlier lists, the observed objects are the top-n outlier lists $\boldsymbol{\sigma}$ from various individual detection methods, and the unknown object is the true top-n outlier list π. The value of the free parameter θ_i depends on the detection performance of the i-th individual method. The goal is to find the optimal ranking π and the corresponding free parameter θ_i which maximize the posteriori probability shown in Eq. 6. In this work, we propose a novel EM algorithm to solve this problem. For obtaining an accurate estimation of θ_i by the EM-based algorithm, we construct the observed objects by applying several queries with different lengths $\{N_q\}_{q=1}^{Q}$, where $N_1 = n$ and $N_{q/1} > n$. Clearly, it is to compute the parameter $\boldsymbol{\theta} = (\theta_1,\cdots,\theta_K)$ by considering the information of different scales. In this paper, the default value of Q is 4 and the lengths meet the following requirement: $N_q = q \cdot n$.

2.4 Inference and Algorithm for OAG

The EM algorithm is widely used for finding the maximum likelihood estimates in the presence of missing data. The procedure includes two steps. First, the expected value of the complete data log-likelihood with respect to the unobserved objects $\phi = \{\pi_q|\pi_q \in R_{N_q}\}_{q=1}^{Q}$, the observed objects $\psi = \{\boldsymbol{\sigma_q}|\boldsymbol{\sigma_q} \in R_{N_q}^K\}_{q=1}^{Q}$, and the current parameter estimate $\boldsymbol{\theta'} = (\theta'_1,\cdots,\theta'_K)$. Second, compute the optimal parameter $\boldsymbol{\theta}$ that maximizes the expectation value in the first procedure. According to the Mallows model and the extended Mallows model, we have the following Lemmas:

Lemma 2: *The expected log-likelihood* $\zeta(\boldsymbol{\theta},\boldsymbol{\theta'})$ *meets the following formula*

$$\zeta(\boldsymbol{\theta},\boldsymbol{\theta'}) = \quad E[\log P(\phi,\psi|\boldsymbol{\theta})|\psi,\boldsymbol{\theta'}] = \sum\nolimits_{(\pi_1,\cdots,\pi_Q)} L(\boldsymbol{\theta})\cdot U(\boldsymbol{\theta'}) \tag{9}$$

where

$$L(\boldsymbol{\theta}) = \sum_{q=1}^{Q} \log P(\pi_q) - \sum_{q=1}^{Q}\sum_{i=1}^{K} \log Z_q(\theta_i) + \sum_{q=1}^{Q}\sum_{i=1}^{K} \theta_i \cdot d(\pi_q, \sigma_q^i) \tag{10}$$

$$U(\boldsymbol{\theta'}) = \prod_{q=1}^{Q} P(\pi_q|\boldsymbol{\theta'}, \boldsymbol{\sigma_q}) \tag{11}$$

Lemma 3: *The parameter $\boldsymbol{\theta}$ maximizing the expected value $\zeta(\boldsymbol{\theta}, \boldsymbol{\theta'})$ meets the following formula:*

$$\sum_{q=1}^{Q} E_{\theta_i}(d(\pi_q, \sigma_q^i)) = \sum_{(\pi_1,\cdots,\pi_Q)} \sum_{q=1}^{Q} d(\pi_q, \sigma_q^i) \cdot U(\boldsymbol{\theta'}) \tag{12}$$

The proofs for Lamma 2 and Lamma 3 are omitted due to lack of space. As shown in Lamma 3, the value of the right-hand side of Eq. 12 and the analytical expression of the left-hand side should be evaluated under the appropriate distance function to obtain the optimal $\boldsymbol{\theta}$. Before introducing the detailed procedure of our EM-based learning algorithm, we bring in an effective distance function $d(\pi, \sigma)$ between the top-n orderings π and σ, which is proposed in [18]. To keep this work self-contained, this distance function is introduced as follows.

Definition 1: *Let F_π and F_σ be the elements of π and σ respectively. $Z = F_\pi \cap F_\sigma$ with $|Z| = z$. $P = F_\pi \setminus Z$, and $S = F_\sigma \setminus Z$ (note that $|P| = |S| = n - z = r$). Define the augmented ranking $\tilde{\pi}$ as π augmented with the elements of S assigned the same index $n + 1$. Clearly, $\tilde{\pi}^{-1}(n + 1)$ is the set of elements at position $n + 1$ ($\tilde{\sigma}$ is defined similarly). Then, $d(\pi, \sigma)$ is the minimum number of the adjacent transpositions needed to turn $\tilde{\pi}$ to $\tilde{\sigma}$ as follows, where $I(x) = 1$ if $x > 0$, and 0 otherwise.*

$$d(\pi, \sigma) = \sum_{\substack{i=1 \\ \tilde{\pi}^{-1}(i)\in Z}}^{n} V_i(\tilde{\pi}, \tilde{\sigma}) + \sum_{\substack{i=1 \\ \tilde{\pi}^{-1}(i)\notin Z}}^{n} U_i(\tilde{\pi}, \tilde{\sigma}) + \frac{r(r + 1)}{2} \tag{13}$$

where

$$V_i(\tilde{\pi}, \tilde{\sigma}) = \sum_{\substack{j=i \\ \tilde{\pi}^{-1}(j)\in Z}}^{n} I(\tilde{\sigma}(\tilde{\pi}^{-1}(i)) - \tilde{\sigma}(\tilde{\pi}^{-1}(j))) + \sum_{j\in\tilde{\pi}^{-1}(n+1)} I(\tilde{\sigma}(\tilde{\pi}^{-1}(i)) - \tilde{\sigma}(j))$$

$$U_i(\tilde{\pi}, \tilde{\sigma}) = \sum_{\substack{j=i \\ \tilde{\pi}^{-1}(j)\in Z}}^{n} 1$$

In each iteration of the EM process, $\boldsymbol{\theta}$ is updated by solving Eq. 12. Based on Definition 1, $E_{\theta_i}(d(\pi_q, \sigma_q^i))$ is computed as follows:

$$E_{\theta_i}(d(\pi_q, \sigma_q^i)) = \frac{N_q e^{\theta_i}}{1 - e^{\theta_i}} - \sum_{j=r+1}^{N_q} \frac{j e^{j\theta_i}}{1 - e^{j\theta_i}} + \frac{r(r + 1)}{2} - r(z + 1)\frac{e^{\theta_i(z+1)}}{1 - e^{\theta_i(z+1)}}$$

This function is a monotonous function of the parameter θ_i. For estimating the right-hand side of Eq. 12, we adopt the Metropolis algorithm introduced in [2] to sample from Eq. 6. Suppose that the current list is π_t. A new list π_{t+1} is achieved by exchanging the objects i and j, which are randomly chosen from all the objects in π_t. Let $r = P(\pi_{t+1}|\boldsymbol{\theta}, \boldsymbol{\sigma})/P(\pi_t|\boldsymbol{\theta}, \boldsymbol{\sigma})$. If $r \geq 1$, π_{t+1} is accepted as the new list, otherwise π_{t+1} is accepted with the probability r. Then, $\boldsymbol{\theta}$ can be computed by the line search approach with the average z of the samples. The steps of *OAG* are shown in Algorithm 2.

Algorithm 2. Order-based aggregation method (*OAG*)

```
Input: $\psi = \{\boldsymbol{\sigma_q}\}_{q=1}^{Q}$ with $|\sigma_q^i| = N_q$, $\boldsymbol{\theta}^{(0)}$, $\varepsilon$, $t = 1, T$
  1. Construct the sampling sets $(\pi_i, \cdots, \pi_Q) \in R_n^Q$ by the Metropolis algorithm from Eq. 6.
  2. Compute the value of the right-hand side of Eq. 12.
  3. Adopt the line search approach to compute $\boldsymbol{\theta}^{(t+1)}$ based on Eq. 12
  4. If $t = T$, or $\sum_{i=1}^{K} |\theta_i^{t+1} - \theta_i^t| < \varepsilon$, return $\boldsymbol{\theta}^{(t+1)}$ and the optimal top-n outlier list $\pi$
     estimated by the sampling procedure; else $t = t + 1$, goto the step 1.
Output: $\boldsymbol{\theta}, \pi$
```

3 Experiments

We evaluate the aggregation performances of *SAG* and *OAG* methods using a number of real world datasets. We measure the robust capabilities of *SAG* and *OAG* methods to the random rankers, which are generated based on the Uniform distribution and the Gaussian distribution, respectively.

3.1 Aggregation on Real Data

In this subsection, we make use of several state-of-the-art methods, including *LOF* [6], *K-Distance* [3], *LOCI* [7], *Active Learning* [10], and *Random Forest* [11] as the individual methods to return the original top-n outliers lists. Since the performances of *LOF* and *K-Distance* depend on the parameter K that determines the scale of the neighborhood, we take the default value of K as 2.5% of the size of a real dataset. Both *LOF* and *LOCI* return outlier scores for each dataset based on the density estimation. However, *K-Distance* [3] only gives objects binary labels. Hence, according to the framework of *K-Distance*, we compute outlier scores as the distance between an object and its Kth nearest neighbor. *Active learning* and *Random Forest* both transform outlier detection to classification based on the artificial outliers generated according to the procedures proposed in [10]. These two methods both compute outlier scores by the majority voting of the weak classifiers or the individual decision trees.

The real datasets used in this section consist of the Mammography dataset, the Ann-thyroid dataset, the Shuttle dataset, and the Coil 2000 dataset, all of which can be downloaded from the UCI database except for the Mammography dataset.[1] Table 1

[1] Thank Professor Nitesh.V.Chawla for providing this dataset, whose email address is *nchawla@nd.edu*

Table 1. Documentations of the real data

Dataset		Mammography	Ann-thyroid	Shuttle-1	Shuttle-2	Shuttle-3	Coil-2000
Number of data	normal	10923	3178	11478	11478	11478	5474
	outlier	260	73	13	39	809	348
Proportion of outliers		2.32%	2.25%	0.11%	0.34%	6.58%	5.98%

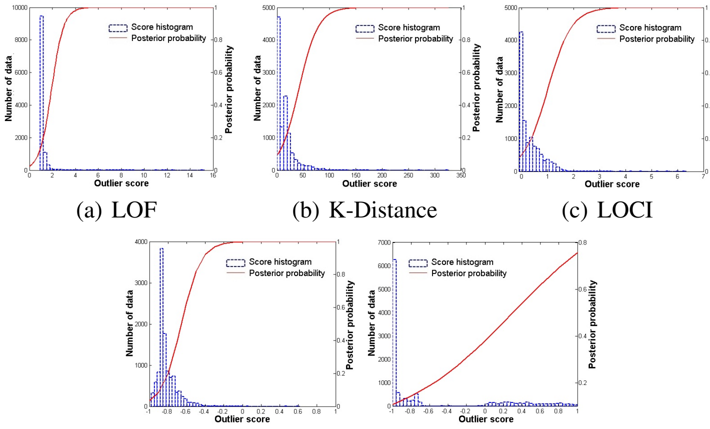
summarizes the documentations of these real datasets. All the comparing outlier detection methods are evaluated using *precision* and *recall* in the top-n outlier list σ as follows

$$Precision = TN/AN \qquad Recall = TN/ON$$

where TN is the number of outliers in ordering σ, AN is the length of σ, and ON is the number of outliers in the dataset. For the quantity AN equals ON in this work, *precision* has the same value with *recall*. Hence, only *precision* is used to measure the performance of each compared method in this section. Clearly, if all the objects in σ are outliers, its *precision* and *recall* both achieve the maximum value 100%. The *Breadth-first* and *Cumulative Sum* methods proposed in *Feature Bagging* [13] are used as the baselines. For *Feature Bagging* does not introduce how to normalize heterogeneous outlier scores, the original outlier scores are processed by the typical normalization method: $S_{norm}(i) = \frac{S(i)-mean}{std}$, where $mean$ is the average score of all the objects and std is the standard deviation of outlier scores. Besides, *Cumulative Sum* requires that every object should be given an outlier score by every individual method. However, for the top-n outlier lists, some objects lying in the ordering σ_i may not be ranked by σ_j. This means that *Cumulative Sum* cannot be applied in the fusion of the top-n outlier lists. Hence, we replace the sum of all the outlier scores with the average of the outlier scores from the individual methods containing the corresponding object for *Cumulative Sum*. The *Mallows Model* [18] is also used as the baseline. As discussed in the previous section, for this algorithm can not combine the basic lists $\boldsymbol{\sigma}$ with various lengths to achieve the true list π, it needs to use all the datasets to compute the expertise indicator parameter θ.

Table 2 lists the experimental results of the individual methods and all the aggregation methods. Figure 2 shows the posterior probability curves based on *SAG* for the individual methods on the Mammography dataset. It is very clear that different detection methods have different scales of outlier scores and posterior probability computed by *SAG* is a monotonic increasing function of outlier scores. In the individual method pool, *LOF* achieves the best performance on the Mammography and the Shuttle-2 datasets, and K-Distance achieves the best performance on the Shuttle-1 dataset. *LOCI* detects the most outliers on the Coil 2000 dataset with *Active learning*. *Random Forest* is superior to the other methods on the Ann-thyroid and Shuttle-3 datasets. However, none of the outliers is detected by *Random Forest* on the Shuttle-1,2 datasets. The above results have verified the motivation that there is a need of ensemble learning in the top-n outlier detection.

From Table 2, we see that *SAG* with $r = 1$ and *SAG* with $r = 0$ achieve the similar performance on all the real datasets. Clearly, for the probability-based *SAG* method,



(a) LOF (b) K-Distance (c) LOCI

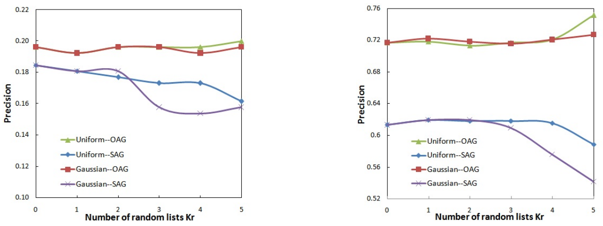
(d) Active Learning (e) Random Forest

Fig. 2. The posterior probability curves based on *SAG* and score histograms of various individual methods on the Mammography dataset

Table 2. The precisions in the top-n outlier lists for all the individual methods and the aggregation methods on the real data

Dataset **Method**	Mammography (Top 260)	Ann-thyroid (Top-73)	Shuttle-1 (Top-13)	Shuttle-2 (Top-39)	Shuttle-3 (Top-809)	Coil-2000 (Top-348)
LOF	19.0%	39.7%	23.1%	53.8%	28.4%	5.5%
K-Distance	13.8%	37.0%	29.8%	48.7%	34.5%	8.0%
LOCI	8.8%	28.8%	7.7%	33.3%	67.0%	8.9%
Active Learning	18.1%	28.8%	15.4%	0%	30.3%	8.9%
Random Forests	15.4%	41.1%	0%	0%	70.6%	8.6%
Average of All	15.0%	35.1%	15.2%	27.2%	46.2%	8.0%
Cumulative Sum	10.0%	31.5%	23.1%	**58.9%**	40.0%	10.3%
Breadth-first	14.2%	38.4%	0%	28.2%	46.9%	10.6%
Mallows Model	13.1%	38.4%	23.1%	51.3%	44.4%	8.0%
SAG (r= 1)	18.5%	34.2%	23.1%	48.7%	61.3%	9.8%
SAG (r= 0)	18.5%	34.2%	23.1%	48.7%	62.1%	9.5%
SAG (r=-1)	5.4%	26.0%	7.7%	43.6%	59.5%	**10.9%**
OAG	**19.7%**	**42.5%**	**30.8%**	53.8%	**71.7%**	9.1%

the number n_d of the individual top-n outlier lists contributes little to the final fusion performance. Compared with the above aggregation methods, the performance of *SAG* with $r = -1$ varies with the nature of the data dramatically. *SAG* with $r = -1$ achieves the best performance on the Coil 200 dataset. However, it performs more poorly than *SAG* with $r = \{1, 0\}$ and *OAG* on the other datasets. This demonstrates that the average of the unified outlier scores does not adapt to the fusion of the top-n lists. In general, since outlier scores are always either meaningless or inaccurate, the order-based aggregation method makes more sense than the score-based method. *OAG* achieves the



(a) Mammography (b) Shuttle-3

Fig. 3. The precisions of *OAG* and *SAG* $(r = 1)$ varying with the number of random lists K_r on the Mammography data and Shuttle-3 data

Table 3. The parameter $\boldsymbol{\theta}$ of all the individual methods and five random lists on the Mammography and Shuttle-3 datasets

Method Dataset		*LOF*	*K-Distance*	*LOCI*	*Active Learning*	*Random Forests*	*random lists (Average)*
Mammogrpahy	*Uniform-Noise*	-0.0058	-0.0039	-0.0058	-0.0052	-0.0039	-0.00014
	Gaussian-Noise	-0.0061	-0.0033	-0.0055	-0.0054	-0.0044	-0.00016
Shuttle-3	*Uniform-Noise*	-0.0014	-0.0016	-0.0014	-0.0018	-0.0037	-0.00001
	Gaussian-Noise	-0.0014	-0.0016	-0.0018	-0.0014	-0.0035	-0.00002

best performance than *SAG* on the Mammography, the Ann-thyroid, and the Shuttle-1,3 datasets. Both *Cumulative Sum* and *SAG* are score-based fusion methods. Table 2 shows that the performance of *SAG* is more stable and effective, especially *SAG* with $r = 1$. *Breath-first*, *Mallows Model*, and *OAG* are all the order-based fusion methods. Although *Breath-first* can be used in the aggregation of top-n outlier lists, it is sensitive to the order of the individual methods. *Mallows Model* supposes that there is a fixed expertise indicator parameter θ for an individual method regardless of the nature of the data. Experiment results indicates that this hypothesis is not appropriate for the ensemble learning in the top-n outlier detection. Overall, *SAG* and *OAG* both achieve the better performances than *Average of All* and the aggregation methods *Breadth-first*, *Cumulative Sum* and *Mallows Model*, which means that the proposed approaches deliver a stable and effective performance independent of different datasets in a good scalability.

3.2 Robustness of Two Aggregation Methods

In this subsection, the goal is to examine the behavior of the *SAG* and *OAG* methods when poor judges are introduced into the individual method pool. For a dataset D, the top-n outlier lists of the poor judges are generated from the underlying distribution U. First, the outlier scores of all the data are sampled from the distribution U. Then, the random top-n outlier lists are obtained by sorting all the data based on the outlier scores. In our experiments, two alternative definitions of U are used: Uniform distribution on the interval $[0, 1]$ and standard Gaussian distribution. The corresponding top-n lists are called *Uniform-Noise* and *Gaussian-Noise*. The individual method pool contains the previous five individual detection methods, and the K_r random lists of the poor judges, where K_r varies from 1 to 5.

For lack of the space, only the results on the Mammography dataset and the Shuttle-3 dataset are shown in the Figure 3. Clearly, *OAG* is more robust to the random poor judges than *SAG* regardless of *Uniform-Noise* or *Gaussian-Noise*. Especially, *OAG* achieves a better performance when the number K_r of random lists increases. Table 3 gives the value of the parameter $\boldsymbol{\theta}$ of the individual method pool on the Mammography and Shuttle-3 datasets. The parameter θ of each *Uniform-Noise* or *Gaussian-Noise* is close to zero. This demonstrates that *OAG* learns to discount the random top-n lists without supervision.

4 Conclusions

We have proposed the general framework of the ensemble learning in the top-n outlier detection in this paper. We have proposed the score-based method (*SAG*) with the normalized method of outlier scores, which is used to transform outlier scores to posterior probabilities. We have proposed the order-based method (*OAG*) based on the distance-based Mallows model to combine the order information of various individual top-n outlier lists. Theoretical analysis and empirical evaluations on several real data sets demonstrate that both *SAG* and *OAG* can effectively combine the state-of-the-art detection methods to deliver a stable and effective performance independent of different datasets in a good scalability, and *OAG* can discount the random top-n outlier lists without supervision.

References

1. Hawkins, D.: Identification of Outliers. Chapman and Hall, London (1980)
2. Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Journal of Biometrika 57(1), 97–109 (1970)
3. Knorr, E.M., Ng, R.T., Tucakov, V.: Distance-based outliers: algorithms and applications. Journal of VLDB 8(3-4), 237–253 (2000)
4. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. Journal of ACM Computing Surveys (CSUR) 31(3), 264–323 (1999)
5. Barnett, V., Lewis, T.: Outliers in Statistic Data. John Wiley, New York (1994)
6. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: Lof: Identifying density-based local outliers. In: SIGMOD, pp. 93–104 (2000)
7. Papadimitriou, S., Kitagawa, H., Gibbons, P.: Loci: Fast outlier detection using the local correlation integral. In: ICDE, pp. 315–326 (2003)
8. Yang, J., Zhong, N., Yao, Y., Wang, J.: Local peculiarity factor and its application in outlier detection. In: KDD, pp. 776–784 (2008)
9. Gao, J., Hu, W., Zhang, Z(M.), Zhang, X., Wu, O.: RKOF: Robust Kernel-Based Local Outlier Detection. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part II. LNCS(LNAI), vol. 6635, pp. 270–283. Springer, Heidelberg (2011)
10. Abe, N., Zadrozny, B., Langford, J.: Outlier detection by active learning. In: KDD, pp. 504–509 (2006)
11. Breiman, L.: Random Forests. J. Machine Learning 45(1), 5–32 (2001)

12. Fox, E., Shaw, J.: Combination of multiple searches. In: The Second Text REtrieval Conference (TREC-2), pp. 243–252 (1994)
13. Lazarevic, A., Kumar, V.: Feature bagging for outlier detection. In: KDD, pp. 157–166 (2005)
14. Gao, J., Tan, P.N.: Converting output scores from outlier detection algorithms into probability estimates. In: ICDM, pp. 212–221 (2006)
15. Nguyen, H., Ang, H., Gopalkrishnan, V.: Mining outliers with ensemble of heterogeneous detectors on random subspaces. Journal of DASFAA 1, 368–383 (2010)
16. Mallows, C.: Non-null ranking models. I. J. Biometrika 44(1/2), 114–130 (1957)
17. Lebanon, G., Lafferty, J.: Cranking: Combining rankings using conditional probability models on permutations. In: ICML, pp. 363–370 (2002)
18. Klementiev, A., Roth, D., Small, K.: Unsupervised rank aggregation with distance-based models. In: ICML, pp. 472–479 (2008)

Towards Personalized Context-Aware Recommendation by Mining Context Logs through Topic Models

Kuifei Yu[1,2], Baoxian Zhang[1], Hengshu Zhu[2,3],
Huanhuan Cao[2], and Jilei Tian[2]

[1] Graduate University of Chinese Academy of Sciences
[2] Nokia Research Center
[3] University of Science and Technology of China
{kuifei.yu,happia.cao,jilei.tian}@nokia.com, bxzhang@gucas.ac.cn,
zhs@ustc.edu.cn

Abstract. The increasing popularity of smart mobile devices and their more and more powerful sensing ability make it possible to capture rich contextual information and personal context-aware preferences of mobile users by user context logs in devices. By leveraging such information, many context-aware services can be provided for mobile users such as personalized context-aware recommendation. However, to the best knowledge of ours, how to mine user context logs for personalized context-aware recommendation is still under-explored. A critical challenge of this problem is that individual user's historical context logs may be too few to mine their context-aware preferences. To this end, in this paper we propose to mine common context-aware preferences from many users' context logs through topic models and represent each user's personal context-aware preferences as a distribution of the mined common context-aware preferences. The experiments on a real-world data set contains 443 mobile users' historical context data and activity records clearly show the approach is effective and outperform baselines in terms of personalized context-aware recommendation.

Keywords: Personalization, Recommender System, Context-Aware, Mobile Users, Latent Dirichlet Allocation (LDA).

1 Introduction

Recent years have witnessed the increasing popularity of smart mobile devices, such as smart phones and pads. These devices are usually equipped with multiple context sensors, such as GPS sensors, 3D accelerometers and optical sensors, which enables them to capture rich contextual information of mobile users and thus support a wide range of context-aware services, including context-aware tour guide [15], location based reminder [13] and context-aware recommendation [2,9,16,10], etc. Moreover, these contextual information and users' corresponding activity (e.g., browsing web sites, playing games and chatting by Social Network Services) can be recorded into *context logs* to be used for mining

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 431–443, 2012.

users' personal context-aware preferences. By considering both context-aware preferences and the current contexts of users, we may be able to make more personalized context-aware recommendations for mobile users.

Indeed, the personalized context-aware recommendations can provide better user experiences than general context-aware recommendations which only take into account contexts but not users' different personal preferences under same contexts. In recent years, many researchers studied the problem of personalized context-aware recommendation [17,12,9]. However, most of this work is based on item ratings generated by users, which are difficult to obtain in practise. In contrast, user activity records in context logs are much easier to get for mobile users.

To the best knowledge of ours, how to mine personal context-aware preferences from context logs and then make personalized context-aware recommendations is still under-explored. To this end, in this paper we attempt to leverage mining user context logs for personalized context-aware recommendation. However, a critical challenge of the problem is that individual user's context logs usually have no sufficient training data for mining personal context-aware preferences. To be specific, as showed in Table 1, it can be observed that many context records have no corresponding activity record. As a result, if we only leverage individual user's context logs for context-aware preference mining, it will be very difficult to learn personal context preferences for recommendation, which is also reflected by our experiments on a real world data set. To address this problem, in this paper, we propose to mine **C**ommon **C**ontext-aware **P**references (CCPs) from many users' context logs through topic models and represent each user's personal context-aware preferences as a distribution of the mined common context-aware preferences. To be specific, first we extract bags of *Atomic Context-Aware Preference (ACP) Features* for each user from their historical context logs. Then, we propose to mine CCPs from users' ACP-feature bags through topic models. Finally, we make recommendations according to the given contexts and CCP distributions of users. Figure 1 illustrates our procedure for generating personalized context-aware recommendation. In addition, we evaluate our proposed approach in a real-world data set of context logs collected from 443 mobile phone users spanning for several months, which contains more than 8.8 million context records, 665 different interactions in 12 content categories.

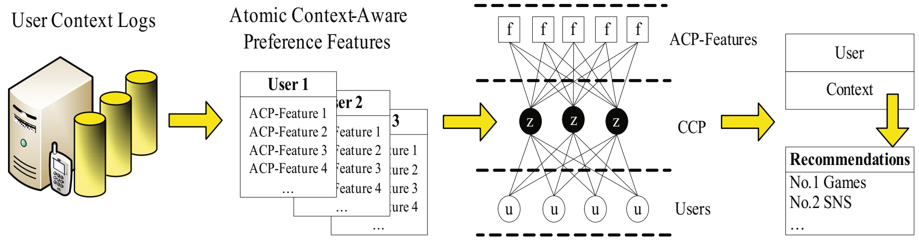


Fig. 1. The procedure of personalized context-aware recommendation for mobile users

Table 1. A toy context log from real-world data set

Timestamp	Context	Activity record
t_1	{(Day name: Monday),(Time range: AM8:00-9:00)), (Profile: General),(Location: Home)}	Null
t_2	{(Day name: Monday),(Time range: AM8:00-9:00)), (Profile: General),(Location: On the way)}	Play action games
t_3	{(Day name: Monday),(Time range: AM8:00-9:00)), (Profile: General),(Location: On the way)}	Null
......		
t_{359}	{(Day name: Monday),(Time range: AM10:00-11:00), (Profile: Meeting),(Location: Work place)}	Null
t_{360}	{(Day name: Monday),(Time range: AM10:00-11:00), (Profile: Meeting),(Location: Work place)}	Browsing sports web sites
......		
t_{448}	{(Day name: Monday),(Time range: AM11:00-12:00), (Profile: General),(Location: Work place)}	Play with Social Network Serivce
t_{449}	{(Day name: Monday),(Time range: AM11:00-12:00), (Profile: General),(Location: Work place)}	Null

The results clearly demonstrate the effectiveness of the proposed approach and indicate some inspiring conclusions.

The remainder of this paper is organized as follows. Section 2 provides a brief overview of related works. Then, Section 3 presents the idea of making personalized context-aware recommendation by mining context logs for mining users' context-aware preferences, and Section 4 presents how to mine common context-aware preferences through topic models. Section 5 reports our experimental results on a real world data set. Finally, in Section 6, we conclude this paper.

2 Related Work

Today, the powerful sensing abilities of smart mobile devices enable to capture the rich contextual information of mobile users, such as location, user activity, audio level, and so on. Consequently, how to leverage such rich contextual information for personalized context-aware recommendation has become a hot problem which dramatically attracts many researchers' attention.

Many previous works about personalized context-ware recommendation for mobile users have been reported. For example, Tung *et al.* [14] have proposed a prototype design for building a personalized recommender system to recommend travel related information according to users' contextual information. Park *et al.* [12] proposed a location-based personalized recommender system, which can reflect users' personal preferences by modeling user contextual information through Bayesian Networks. Bader *et al.* [2] have proposed a novel context-aware approach to recommending points-of-interest (POI) for users in an automotive scenario. Specifically, they studied the scenario of recommending gas stations for car drivers by leveraging a Multi-Criteria Decision Making (MCDM) based methods to modeling context and different routes. However, most of these works only leverage individual user's historical context data for modeling personal context-aware preferences, and do not take into account the problem of insufficient personal training data.

Actually, the problem of insufficient personal training data is common in practice and many researchers have studied how to address this problem. For example, Woerndl *et al.* [16] proposed a hybrid framework named "play.tools" for recommending mobile applications by leveraging users' context information. This recommendation framework are based on what other users have installed in similar context will be liked by a given user. Kim *et al.* [9] investigated several Collaborative Filtering (CF) based approaches for recommendation and developed a memory based CF approach to providing context-aware advertisement recommendation. Specially, the proposed approach can leverage a classification rule of decision tree to understand users' personal preference. Zheng *et al.* [17] have studied a model based CF approach to recommending user locations and activities according to users' GPS trajectories. The approach can model user, location and activity as a 3-dimensional matrix, namely tensor, and perform tensor factorization with several constraints to capture users' preferences. Alexandros *et al* [10] proposed a model based CF approach for making recommendation with respect to rich contextual information, namely multiverse recommendation. Specifically, they modeled the rich contextual information with item by N-dimensional tensor, and proposed a novel algorithm to make tensor factorization. In a word, most of these approaches are based on rating logs of mobile users and the objective is to predict accurate ratings for the unobserved items under different contexts. However, usually we cannot obtain such rating data in user mobile devices. In contrast, it is easier to collect context logs which contain users' historical context data and activity records, which motivates our work for exploring how to leverage context logs for personalized context-aware recommendation.

The proposed approach in this paper exploits topic models for learning users' CCPs. Indeed, topic models are widely used in text retrieval and information extraction. Typical topic models include the Mixture Unigram (MU) [11], the Probabilistic Latent Semantic Indexing (PLSI) [8], and the Latent Dirichlet Allocation (LDA) [4]. Most of other topic models are extended from the above ones for satisfying some specific requirements. In our approach, we exploit the widely used LDA model.

3 Preliminary

As mentioned in Section 1, smart devices can capture the historical context data and corresponding activity records of users through multiple sensors and record them in context logs. For example, Table 1 shows a toy context log of a mobile user, which contains several *context records*, and each context record consists of a timestamp, the most detailed available context at that time, and the corresponding user activity record captured by devices. A context consists of several *contextual features* (e.g., Day name, Time range, and Location) and their corresponding values (e.g., Saturday, AM8:00-9:00, and Home), which can be annotated as *contextual feature-value pairs*. And we mention "available" because a context record may miss some context data though which context data which should be collected is usually predefined. For example, the GPS coordinate is not available when the user is indoor. Moreover, interaction records can be empty (denoted as “Null”) because the user activities which can be captured by devices do not always happen.

It is worth noting that we transform raw location based context data such as GPS coordinates or cell Ids into social locations which have explicit meanings such as “Home” and “Work place” by some existing location mining approaches (e.g., [5]). The basic idea of these approaches is to find clusters of user location data and recognize their social meaning by time pattern analysis. Moreover, we also manually transform the raw activity records to more general ones by mapping the activity of using a particular application or playing a particular game to an activity category. For example, we can transform two raw activity records “*Play Angry Birds*” and “*Play Fruit Ninja*” to same activity records “*Play action games*”. In this way, the context data and activity records in context logs are normalized and the data sparseness is some how alleviated for easing context-aware preference mining.

Given a context $C = \{p\}$ where p denotes an atomic context, i.e., a contextual feature-value pair, the probability that a user u prefers activity a can be represented as

$$P(a|C,u) = \frac{P(a,C|u)P(u)}{P(C,u)} \propto P(a,C|u) \propto \prod_p P(a,p|u),$$

where we assume that the atomic contexts are mutually conditionally independent given u.

Then the problem becomes how to calculate $P(a,p|u)$. According to our procedure, we introduce a variable of CCP denoted as z, and thus we have

$$P(a,p|u) = \sum_z P(a,p,z|u) \propto \sum_z P(a,p|z,u)P(z,u) \propto \sum_z P(a,p|z)P(z|u),$$

where we assume that a user's preference under a context only relies on the CCPs and his (her) context-aware preferences in the form of their distribution on the CCPs, rather than other information of the user. Therefore, the problem is further converted into learning $P(a,p|z)$ and $P(z|u)$ from many users' context logs, which can be solved by widely used topic models. In the next section, we present how to utilize topic models for mining CCPs, i.e., $P(a,p|z)$, and accordingly make personalized context-aware recommendation.

4 Mining Common Context-Aware Preferences through Topic Models

Topic models are generative models that are successfully used for document modeling. They assume that there exist several topics for a corpus D and a document d_i in D can be taken as a bag of words $\{w_{i,j}\}$ which are generated by these topics. For simplicity, we refer the co-occurrence of a user activity a

and the corresponding contextual feature-value pair p, i.e., (a, p), as *Atomic Context-aware Preference feature*, and *ACP-feature* for short. Intuitively, if we take ACP-features as words, take context logs as bags of ACP-features to correspond documents, and take CCPs as topics, we can take advantage of topic models to learn CCPs from many users' context logs.

However, raw context logs are not naturally in the form of bag of ACP-features so we need some preprocessing for extracting training data. Specially, we first remove all context records without any activity record and then extract ACP-feature from the remaining ones. Given a context record $< Tid, C, a >$ where Tid denotes the timestamp, $C = \{p_1, p_2, ..., p_l\}$ denotes the context and a denotes the activity, we can extract l ACP-features, namely, (a, p_1), (a, p_2), ..., (a, p_l). For simplicity, we refer the bag of ACP-features extracted from user u's context log as the *ACP-feature bag* of u.

Among several existing topic models, in this paper, we leverage the widely used Latent Dirichlet Allocation model (LDA) [4]. According to LDA model, the ACP-feature bag of user u_i denoted as d_i is generated as follows. First, before generating any ACP-feature bag, K prior ACP-feature conditional distributions given context-aware preferences $\{\phi_z\}$ are generated from a prior Dirichlet distribution β. Secondly, a prior context-aware preference distribution θ_i is generated from a prior Dirichlet distribution α for each user u_i. Then, for generating the j-th ACP-feature in d_i denoted as $w_{i,j}$, the model firstly generates a CCP z from θ_i and then generates $w_{i,j}$ from ϕ_z. Figure 2 shows the graphic representation of modeling ACP-feature bags by LDA.

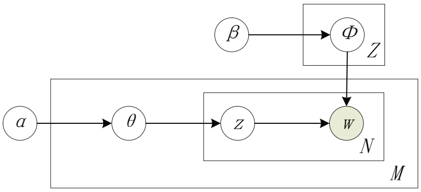


Fig. 2. The graphical model of LDA

In our approach, the objective of LDA model training is to learn proper estimations for latent variables θ and ϕ to maximize the posterior distribution of the observed ACP-feature bags. In this paper, we choose a Markov chain Monte Carlo method named Gibbs sampling introduced in [6] for training LDA models efficiently. This method begins with a random assignment of CCPs to ACP-features for initializing the state of Markov chain. In each of the following iterations, the method will re-estimate the conditional probability of assigning a CCP to each ACP-feature, which is conditional on the assignment of all other ACP-features. Then a new assignment of CCP to ACP-features according to those latest calculated conditional probabilities will be scored as a new state of Markov chain. Finally, after rounds of iterations, the assignment will converge, which means each ACP-feature is assigned a stable and final CCP. Eventually, we can obtain the estimated values for two distributions $\{\widetilde{p}(a, p|z)\}$ and $\{\widetilde{p}(z|u)\}$, which denote the probability that the ACP-feature (a, p) appears under the CCP z, and the probability that user u has the context-aware preference z, respectively.

$$\widetilde{p}(a, p|z) = \frac{n_{(z)}^{(a,p)} + \beta}{n_{(a,p)}^{(\cdot)} + A\beta}, \quad \widetilde{p}(z|u) = \frac{n_{(z)}^{(u)} + \alpha}{n_{(\cdot)}^{(u)} + Z\alpha},$$

where the $n_{(z)}^{(a,p)}$ indicates the number of times ACP-feature (a, p) has been assigned to CCP z, while $n_{(z)}^{(u)}$ indicates the number of times a ACP-feature from user u's context log that has been assigned to CCP z. The A indicates the number of ACP-features from u's context log, and Z indicates the number of CCPs.

LDA model needs a predefined parameter Z to indicate the number of CCPs. How to select an appropriate Z for LDA is an open question. In terms of guaranteeing the performance of recommendation, in this paper we utilize the method proposed by Bao et al [3] to estimate Z, and we set ζ to be 10% in our experiments accordingly. Please refer to [1] for more information.

After learning CCPs represented by distributions of ACP-features, we can predict users' preference according to their historical context-aware preferences and current contexts, i.e., $P(a, C|u)$. Then, we recommend users a rank list of different categories of contents according to the preference prediction. For example, if we predict a user u is more likely willing to play action games than listen pop music, the recommendation priority of popular action games will be higher than that of recent hot pop music.

5 Experiments

In this section, we evaluate the performance of our LDA based personalized context-aware recommendation approach, namely **P**ersonalized **C**ontext-aware **R**ecommendation with **LDA** (PCR-LDA), with several baseline methods in a real-world data set.

5.1 Data Set

The data set used in the experiments is collected from many volunteers by a major manufacturer of smart mobile devices. The data set contains context logs with rich contextual information and user activities of 443 smart phone users spanning for several months. The detailed statistics of our data set are illustrated in Table 2. From table 2 we can observe that only 12.5% context records have activities, which indicates the insufficient activity records for individual user in practice. Moreover, Table 3 shows the concrete types of context data contained in our data set. In addition, in our data set, all activities can be classified into 12

Table 2. Statistics of our data set

	Number
users	443
unique activities	665
unique context	4,391
context records	8,852,187
activity-context records†	1,097,189

† activity-context records denote the context records with non-empty user activity records.

content categories, which are *Call*, *Web*, *Multimedia*, *Management*, *Games*, *System*, *Navigation*, *Business*, *Reference*, ***S**ocial **N**etwork **S**ervice (SNS)*, *Utility* and *Others*. Specifically, in our experiments, we do not utilize the categories *Call* and *others* because their activity information is clear for making recommendations. Therefore, in our experiments we utilize 10 activity categories which contain 618 activities appear in total 408,299 activity-context records.

Table 3. The types of contextual information in our data set

Data source	Data type	Value range
Time Info	Week	{Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday}
	Is a holiday?	{Yes, No}
	Day period	{Morning(AM7:00-AM11:00), Noon(AM11:00-PM14:00), Afternoon(PM14:00-PM18:00), Evening(PM18:00-PM21:00), Night(PM21:00-Next day AM7:00)}
	Time range	{AM0:00-AM1:00, AM1:00-AM2:00, AM2:00-AM3:00, ... , PM23:00-PM24:00}
System Info	Profile type	{General, Silent, Meeting, Outdoor, Pager, Offline}
Geo Info	Location	{Home, Work Place, On the way}.

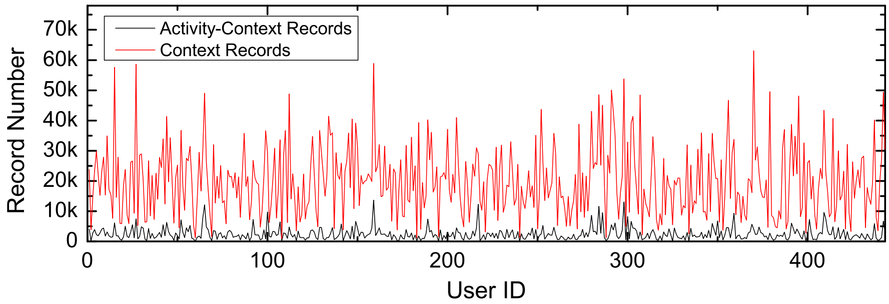


Fig. 3. The distribution of context coverage for all users

Figure 3 shows the distribution of context records and activity-context records for all users. From the figure we can see that usually though the context records of individual mobile users are sufficient, only small proportion of them have non-empty activity records and can be used as training data, which implies the limit of learning personal context-aware preferences only from individual user's context logs.

5.2 Benchmark Methods

To evaluate the recommendation performance of our approach, we chose two context-aware baseline methods as follows.

CPR stands for Context-aware Popularity based Recommendation which is a basic context-aware recommendation approach without considering personal context-aware preference. To be specific, in this approach, given a user u and a context C, we predict user preferred activities by the most frequent activities appear under C according to all users' historical context logs and recommend corresponding contents. This popularity based approach is widely used in practical recommender systems.

PCR-i stands for Personalized Context-aware Recommendation by only leveraging Individual user's context logs. To be specific, in this approach, given a user u and a context C, we rank each activity a by probability $P(a|u,C)$, which can be estimated by $P(a|u,C) \propto (\prod_{p \in C} P(a,p|u)$. The probability $P(a,p|u)$ can be calculated by $P(a,p|u) = \frac{n_{a,p}}{n_{(.)}}$, where $n_{a,p}$ and $n_{(.)}$ indicate the numbers of ACP-feature (a,p) and all ACP-features appeared in the context log of u, respectively.

5.3 Evaluation Metrics

In the experiments, we utilize 5-fold-cross validation to evaluate the performance of each recommendation approaches. To be specific, we first randomly divide each user's context log into five equal parts, then use each part as test data while use other four parts as training data for total 5-rounds of recommendation. In the test process, we only take into account the context records with non-empty activity records, and use the contexts and the content categories corresponding to the real user activity as context input and ground truth, respectively. In our experiments, each recommendation approach will return a ranked list of recommended content categories according to predicted user activities. To evaluate the performance of each approach, we leverage two different metrics as follows.

MAP@K stands for Mean Average Precision at top K recommendation results. To be specific, $MAP@K = \frac{\sum AP^{(u)}@K}{|U|}$, where $AP^{(u)}@K$ denotes the average precision at top k recommendation results on the test cases of user u, and $|U|$ indicates the number of the users. $AP^{(u)}@K$ can be computed by $\frac{1}{N_u} \sum_i \sum_{r=1}^{K} (P_i(r) \times rel_i(r))$, where N_u denotes the number of test cases for user u, r denotes a given cut-off rank, $P_i(r)$ denotes the precision on the i-th test case of u at a given cut-off rank r, and $rel_i()$ is the binary function on the relevance of a given rank.

MAR@K stands for Mean Average Recall at top K recommendation results. To be specific, $MAR@K = \frac{\sum AR^{(u)}@K}{|U|}$, where $AR^{(u)}@K$ denotes the average recall at top k recommendation results on the test cases of user u, and $|U|$ indicates the number of the users. $AR^{(u)}@K$ can be computed by $\frac{1}{N_u}\sum_i \sum_{r=1}^{K} rel_i(r)$, where N_u denotes the number of test cases for user u, r denotes a given cut-off rank, and $rel_i()$ is the binary function on the relevance of a given rank.

5.4 Overall Results of Recommendation

To evaluate our PCR-LDA recommendation approach, we compare its recommendation performance with other baselines. To be specific, according to the parameter estimation approaches introduced in Section 4, the number of CCPs for LDA training is set to be 15. In Section 5.5 we will further discuss the setting of this parameter. For the LDA training, the two parameters α and β are empirically set to be $50/Z$ and 0.2 according to discussion in [7]. Both PCR-LDA and the two baselines are implemented by C++ and the experiments are conducted on a 3GHZ×4 quad-core CPU, 3G main memory PC.

We first test the $MAP@K$ performance of each recommendation approach with respect to varying K, which are shown in Figure 4. From the results we can observe that PCR-LDA outperforms other baselines with a significant margin.

Figure 5 shows the $MAR@K$ of each recommendation approach. From the results we can observe our PCR-LDA can achieve 100% performance when $K = 10$, which means they can return recommendation list contains at least one ground

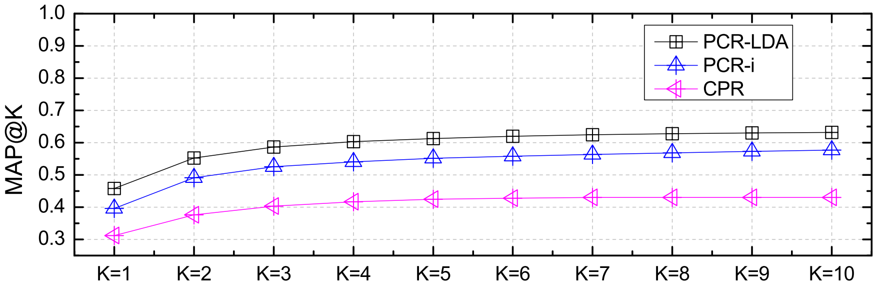


Fig. 4. The MAP@K performance of each recommendation approach

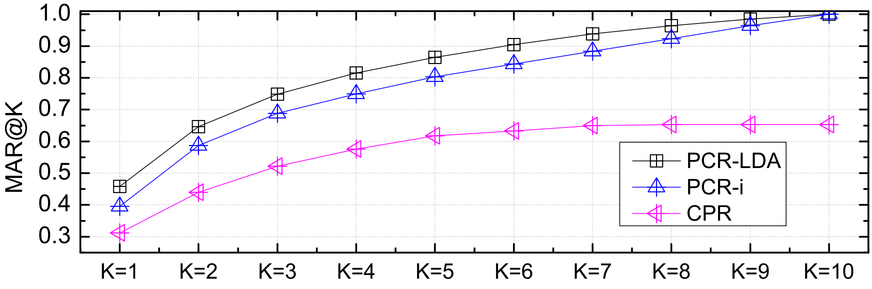


Fig. 5. The MAR@K performance of each recommendation approach

truth activities for all contexts. It is because PCR-LDA takes advantage of many users' context logs. In contrast, PCR-i has worse MAR@K due to the insufficient training data in individual user's context logs for mining context-aware preference. Moreover, due to the different context-aware preference between users, the popularity based approach CPR under-performs the other approaches.

5.5 Robustness Analysis

CPR-LDA needs a parameter Z to determine the number of CCPs. Although we can empirically select Z by estimating perplexity, we still study the impact of such parameter to our recommendation results. Figure 6 shows the MAP@10 of PCR-LDA with respect to varying Z. From the results we can observe that the MAP@10 of PCR-LDA is impacted dramatically by a relatively small Z and becomes stable with relatively big Z. It is because when a relatively small Z is selected, all ACP-features may have strong relationships with each CCP. Thus the approach is actually near to combine all users' context logs as one log for recommendation, which will introduce many noisy data. Another interesting phenomenon is that the MAP@10 peaks when Z is set to be 15, which is consistent with our experimental setting and implies the parameter selection method if effective. The experimental results of $MAP@K$ with other settings of K show the similar phenomena.

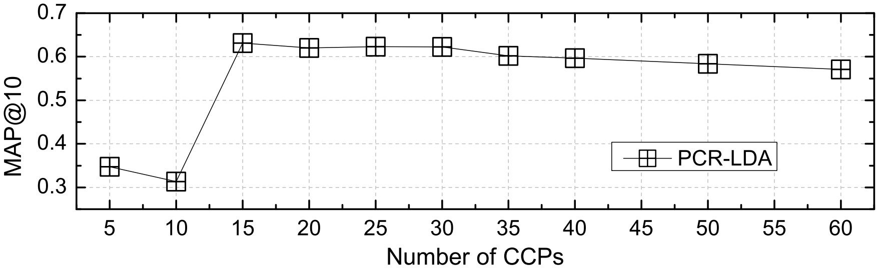


Fig. 6. The MAP@10 performance of PCR-LDA to varying number of CCPs

5.6 Case Study

In addition to the studies on the overall performance of our recommendation approach, we also study the cases in which PCR-LDA outperforms the baselines. For example, Table 4 shows the top 3 recommendation results of each approach for two test cases of two different users given the "{(Is holiday: No), (Day period: Evening), (Time range: PM22:00-23:00), (Day name: Monday), (Profile: General), (Location: Home)}", which may imply the users' leisure time at home. In this case, the activity records of user #152 and user #343's test cases are *Multimedia* and *Web*, respectively. From the results, we can observe that PCR-LDA recommend relevant content categories in the top one position. In contract, PCR-i can only recommend relevant content categories in the top one position for one test case, and the Popularity based approach CPR always recommend same content categories for all users and thus sometimes performs not well.

Table 4. An example of recommendation results for user #152 and #343

Context	{(Time range: PM22:00-23:00),(Is holiday: No),(Day name: Monday),(Day period: Night),(Profile: Offline),(Location: Home)}

Top 3 Recommendation Results for user *#152*	
Ground truth	*Multimedia*
PCR-LDA	Multimedia ($\surd$), Web, Game
PCR-i	Multimedia ($\surd$), Business, Management
CPR	Web, System, Business

Top 3 Recommendation Results for user *#343*	
Ground truth	*Web*
PCR-LDA	Web ($\surd$), Multimedia, SNS
PCR-i	Multimedia, Game, Web
CPR	Web ($\surd$), System, Business

6 Concluding Remarks

In this paper, we investigated how to exploit user context logs for personalized context-aware recommendation by mining CCPs through topic models. To be specific, first we extract ACP-Feature bags for each user from their historical context logs. Then, we propose to mine users' CCPs through topic models. Finally, we make recommendation according to the given context and the CCP distribution of the given user. The experimental results from a real-world data set clearly show that our proposed recommendation approach can achieve good performance for personalized context-aware recommendation.

References

1. Azzopardi, L., Girolami, M., Risjbergen, K.V.: Investigating the relationship between language model perplexity and ir precision-recall measures. In: SIGIR 2003, pp. 369–370 (2003)
2. Bader, R., Neufeld, E., Woerndl, W., Prinz, V.: Context-aware poi recommendations in an automotive scenario using multi-criteria decision making methods. In: CaRR 2011, pp. 23–30 (2011)
3. Bao, T., Cao, H., Chen, E., Tian, J., Xiong, H.: An unsupervised approach to modeling personalized contexts of mobile users. In: ICDM 2010, pp. 38–47 (2010)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Lantent dirichlet allocation. Journal of Machine Learning Research, 993–1022 (2003)
5. Eagle, N., Clauset, A., Quinn, J.A.: Location segmentation, inference and prediction for anticipatory computing. In: AAAI Spring Symposium on Technosocial Predictive Analytics (2009)
6. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of National Academy of Science of the USA, 5228–5235 (2004)
7. Heinrich, G.: Paramter stimaion for text analysis. Technical report, University of Lipzig (2009)

8. Hofmann, T.: Probabilistic latent semantic indexing. In: SIGIR 1999, pp. 50–57 (1999)
9. Jae Kim, K., Ahn, H., Jeong, S.: Context-aware recommender systems using data mining techniques. Journal of World Academy of Science, Engineering and Technology 64, 357–362 (2010)
10. Karatzoglou, A., Amatriain, X., Baltrunas, L., Oliver, N.: Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In: RecSys 2010, pp. 79–86 (2010)
11. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Machine Learning 39, 103–134 (2000)
12. Park, M.-H., Hong, J.-H., Cho, S.-B.: Location-Based Recommendation System Using Bayesian User's Preference Model in Mobile Devices. In: Indulska, J., Ma, J., Yang, L.T., Ungerer, T., Cao, J. (eds.) UIC 2007. LNCS, vol. 4611, pp. 1130–1139. Springer, Heidelberg (2007)
13. Sohn, T., Li, K.A., Lee, G., Smith, I., Scott, J., Griswold, W.G.: Place-Its: A Study of Location-Based Reminders on Mobile Phones. In: Beigl, M., Intille, S.S., Rekimoto, J., Tokuda, H. (eds.) UbiComp 2005. LNCS, vol. 3660, pp. 232–250. Springer, Heidelberg (2005)
14. Tung, H.-W., Soo, V.-W.: A personalized restaurant recommender agent for mobile e-service. In: EEE 2004, pp. 259–262 (2004)
15. van Setten, M., Pokraev, S., Koolwaaij, J.: Context-Aware Recommendations in the Mobile Tourist Application COMPASS. In: De Bra, P.M.E., Nejdl, W. (eds.) AH 2004. LNCS, vol. 3137, pp. 235–244. Springer, Heidelberg (2004)
16. Woerndl, W., Schueller, C., Wojtech, R.: A hybrid recommender system for context-aware recommendations of mobile applications. In: ICDE 2007, pp. 871–878 (2007)
17. Zheng, V.W., Cao, B., Zheng, Y., Xie, X., Yang, Q.: Collaborative filtering meets mobile recommendation: A user-centered approach. In: AAAI 2010, pp. 236–241 (2010)

Mining of Temporal Coherent Subspace Clusters in Multivariate Time Series Databases

Hardy Kremer, Stephan Günnemann, Arne Held, and Thomas Seidl

RWTH Aachen University, Germany
{kremer,guennemann,held,seidl}@cs.rwth-aachen.de

Abstract. Mining temporal multivariate data by clustering techniques is recently gaining importance. However, the temporal data obtained in many of today's applications is often complex in the sense that interesting patterns are neither bound to the whole dimensional nor temporal extent of the data domain. Under these conditions, patterns mined by existing multivariate time series clustering and temporal subspace clustering techniques cannot correctly reflect the true patterns in the data.

In this paper, we propose a novel clustering method that mines temporal coherent subspace clusters. In our model, these clusters are reflected by sets of objects and relevant intervals. Relevant intervals indicate those points in time in which the clustered time series show a high similarity. In our model, each dimension has an *individual* set of relevant intervals, which *together* ensure temporal coherence. In the experimental evaluation we demonstrate the effectiveness of our method in comparison to related approaches.

1 Introduction

Mining patterns from multivariate temporal data is important in many applications, as for example analysis of human action patterns [12], gene expression data [8], or chemical reactions [17]. Temporal data in general reflect the possibly changing state of an observed system over time and are obtained by sensor readings or by complex simulations. Examples include financial ratios, engine readings in the automotive industry, patient monitoring, gene expression data, sensors for forest fire detection, and scientific simulation data, as e.g. climate models. The observed objects in these examples are individual stocks, engines, patients, genes, spatial locations or grid cells in the simulations. The obtained data are usually represented by multivariate time series, where each attribute represents a distinct aspect of observed objects; e.g., in the health care example, each patient has a heart rate, a body temperature, and a blood pressure. The attributes are often correlated; e.g. for the forest fire, the attributes temperature and degree of smoke are both signs of fire. Unknown patterns in such databases can be mined by clustering approaches, where time series are grouped together by their similarity. Accordingly, clusters of time series correspond to groups of objects having a similar evolution over time, and clusters represent these evolutions.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 444–455, 2012.

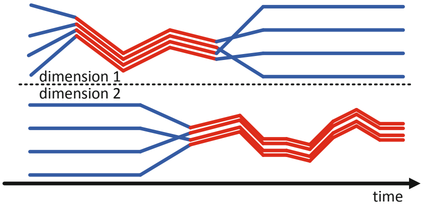


Fig. 1. Multivariate temporal pattern with two intervals

In many applications, however, existing approaches for clustering univariate or multivariate time series are ineffective due to a specific aspect of the analyzed data: patterns of interest that are neither bound to the whole dimensional nor temporal extent of the time series. Since our method is designed for effective mining under this scenario, we elaborate on this aspect in the next paragraph.

Temporal patterns of interest often only exist over a partial temporal extent of analyzed time series, i.e. they are constrained to an interval, and a single multivariate temporal pattern can have different intervals for each of its dimensions, as it is illustrated in Fig. 1. More concretely, time series belonging to one cluster only have similar values in these intervals, and values in the remaining intervals are noisy. Also, for some clusters there are dimensions in which there is no similarity between the time series. In the following, the intervals and dimensions belonging to a cluster are called relevant, while the remaining intervals and dimensions are called non-relevant. If non-relevant intervals are considered in distance measures that are used to decide which time series are grouped together, clusters can result that do not reflect the true patterns in the data.

Our novel, subspace clustering related approach handles this aspect by using an effective cluster model that distinguishes explicitly between relevant and non-relevant intervals for each cluster, and only relevant intervals are incorporated into the similarity function used for deciding which time series belong to a specific cluster. Our approach prevents incoherent time series clusters, i.e. clusters that have points in time that do not belong to any of the cluster's individual intervals. This ensures that there are no single (incoherent) cluster which would better be represented by several single (coherent) clusters.

Summarized, we propose a novel, subspace clustering related approach for effective clustering of multivariate time series databases that

- uses a cluster definition that distinguishes explicitly between relevant and non-relevant intervals for each individual dimension; the individual intervals as a whole form a temporal coherent cluster.
- is efficient due a approximate computation of our model that delivers high quality results.

This paper is structured as follows: in Section 2 we discuss related work. In Section 3 we introduce our approach for effective clustering of multivariate time series, for which an efficient algorithm is presented in Section 4. In Section 5 we evaluate our approach and Section 6 contains concluding remarks.

2 Related Work

Clustering of temporal data can roughly be divided into clustering of incoming data stream, called stream clustering [9], and clustering of static databases, the topic of this paper. Our method is related to two static clustering research areas, namely time series clustering and subspace clustering, and we discuss these areas in the following. In the experiments, we compare to methods from both areas.

Time Series Clustering. There is much research on clustering *univariate* time series data, and we suggest the comprehensive surveys on this topic [4,11]. Clustering of *multivariate* time series is recently gaining importance: Early work in [14] uses clustering to identify outliers in multivariate time series databases. Multivariate time series clustering approaches based on statistical features were introduced in [2,19,20]. There are no concepts in these approaches to discover patterns hidden in parts of the dimensional or temporal extents of time series.

Most clustering approaches are based on an underlying *similarity measure* between time series. There is work on noise-robust similarity measures based on partial comparison, called Longest Common Subsequences (LCSS) [18]. We combined k-Medoid with LCSS as a competing solution.

There is another type of time series clustering methods, designed for applications in which single, long time series are mined for frequently appearing subsequences. These methods perform *subsequence clustering*, with subsequences being generated by a sliding window. Since we are interested in patterns that occur in several time series at similar points in time and not in patterns that occur in a single time series at arbitrary positions, those approaches cannot be applied in our application scenario.

Subspace Clustering (2D) and TriClustering (3D). Subspace clustering [1,10,15,21] was introduced for high-dimensional (non-temporal) vector data, where clusters are hidden in individual dimensional subsets of the data. Since subspace clustering is achieved by simultaneous clustering of the objects and dimensions of dataset, it is also known as 2D clustering. When subspace clustering is applied to 3D data (objects, dimensions, time points), the time series for the individual dimensions are concatenated to obtain a 2D space (objects, concatenated dimensions). While subspace clustering is good for excluding irrelevant points in time, there are problems when it is applied to temporal data: First, by the transformation described above, the correlation between the dimensions is lost. Second, subspace clustering in general cannot exploit the natural correlation between subsequent points in time, i.e. temporal coherence is lost.

Accordingly, for 3D data, *Triclustering* approaches were introduced [7,8,16,22], which simultaneously cluster objects, dimensions, and points in time. Special Triclustering approaches are for clustering two related datasets together [6], which is a fundamentally different concept than the one in this paper. Generally, Triclustering approaches can only find block-shaped clusters: A cluster is defined by a set of objects, dimensions, and points in time [22] or intervals [7,16]. The points in time or intervals hold for all objects and dimensions of a cluster.

In contrast, our approach can mine clusters where each dimension has different, independent relevant intervals.

3 A Model for Effective Subspace Clustering of Multivariate Time Series Data

In the following we introduce our model for subspace clustering of multivariate time series data. In Section 3.1, we introduce our definition for subspace clusters of complex multivariate time series data, and in Section 3.2 we formalize an optimal clustering that is redundancy-free and contains clusters of maximal interestingness.

3.1 Time Series Subspace Cluster Definition

As input for our model we assume a database DB of multivariate time series where $Dim = \{1, \ldots, Dim_{max}\}$ denotes the set of dimensions and $T = \{1, \ldots, T_{max}\}$ the set of points in time for each time series. We use $o[d,t] \in \mathbb{R}$ to refer to the attribute value of time series $o \in DB$ in dimension $d \in Dim$ at time $t \in T$. As an abbreviation, $o[d, t_1 \ldots t_2]$ denotes the univariate subsequence obtained from object o by just considering dimension d between points in time t_1 and t_2. Our aim is to detect temporal coherent patterns, i.e. similar behaving objects, in this kind of multivariate data.

Since we cannot expect to find temporal patterns over the whole extent of the time series or within all dimensions, we have to restrict our considerations to subsets of the overall domain. Naively, a cluster could be defined by a tuple (O, S, I) where the objects O show similar behavior in subspace $S \subseteq Dim$ and time points $I \subseteq T$. This is straightforward extension of subspace clustering to the temporal domain and is used in triclustering approaches like [7,16,22].

A model based on this extension is limited because each selected dimension $d \in S$ has to be relevant for each selected point in time $t \in I$. For example, if the objects O are similar in d_1 at time t_1 and in d_2 at t_2 but not similar in d_1 at time t_2, we cannot get a single cluster for the objects O. We either have to exclude dimension d_1 or time point t_2 from the cluster. Thus, important information is lost and clustering effectiveness degrades.

Our novel model avoids this problem by selecting per dimension an individual set of intervals in which the time series are similar in (cf. Fig. 1). Such an interval, which contains a specific temporal pattern, is denoted as interval pattern.

Definition 1. *An **interval pattern** $IP = (O, d, Int)$ is defined by:*

- *an object set $O \subseteq DB$*
- *one selected dimension $d \in Dim$*
- *an interval $Int = \{start, \ldots, end\} \subseteq T$ with $length(Int) > 1$ for $length(Int) := end - start + 1$, i.e. we only permit non-trivial intervals.*
- *a specific **cluster property** the corresponding subsequences $o[d, start \ldots end]$ with $o \in O$ have to fulfill. We use the compactness of clusters based on the Maximum Norm: i.e., $\forall o, p \in O \; \forall t \in Int : \; |o[d,t] - p[d,t]| \leq w$*

To avoid isolated points in time, i.e. where time series are rather similar by chance, we require that $length(Int) > 1$. The cluster property defines how similarity between subsequences is measured and how similar they need to be in order to be included in the same interval pattern. This property can be chosen by specific application needs. Besides the cluster compactness, which is also used by other subspace clustering methods [13,15], other distance measures applicable for time series including DTW [3] can be used.

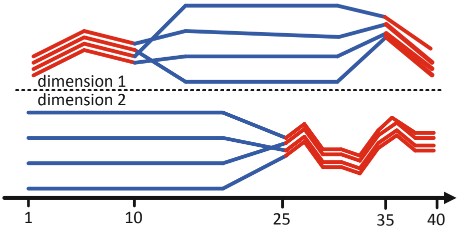


Fig. 2. Example for incoherent patterns

Based on the introduced interval patterns, clusters are generated: for each dimension, zero, a single, or even several interval patterns can exist in the cluster. This allows our method to systematically exclude non-relevant intervals from the cluster to better reflect the existing patterns in the analyzed data. However, not all combinations of interval patterns correspond to reasonable temporal clusters. The temporal coherence of the pattern is crucial. For example, let us consider a set of objects O forming three interval patterns in the time periods 1-10, 25-40, and 35-40, as illustrated in Fig. 2. Since for the remaining points in time no pattern is detected, there is no temporal coherence of the patterns. In this case, two individual clusters would reflect the data correctly.

To ensure temporal coherence, each point in time $t \in T$ that is located between the beginning $a \in T$ and ending $b \in T$ of a cluster, i.e. $a \leq t \leq b$, has to be contained in at least one interval pattern of an arbitrary dimension. Thus, by considering all dimensions simultaneously, the cluster has to form a single connected interval, and each point in time can be included in several dimensions.

Definition 2. ***TimeSC***. *A coherent time series subspace cluster (TimeSC) $C = (O, \{(d_i, Int_i)_{\{1\ldots m\}}\})$, i.e. an object set together with intervals in specific dimensions, is defined by:*

- *for each interval $i \in \{1, \ldots, m\}$ it holds that (O, d_i, Int_i) is a valid interval pattern.*
- *the intervals per dimension are disjoint, i.e.* $\forall i, j \in \{1, \ldots, m\}, i \neq j : \; Int_i \cap Int_j = \emptyset \vee d_i \neq d_j$
- *the cluster is temporal coherent, i.e. combined, we have a single connected interval:* $\exists a, b \in T, a \leq b : \; \bigcup_{i=1}^{m} Int_i = \{a, \ldots, b\}$.

We require disjoint intervals per dimension because for overlapping intervals single points in time could be included multiple times in the cluster, which is obviously not beneficial for describing the cluster.

Overall, our novel cluster model avoids the drawbacks of previous methods and flexibly identifies the coherent temporal patterns in complex multivariate time series data.

3.2 Clustering Model: Redundancy Avoidance

Accounting for the properties of temporal data, an object can naturally occur in several clusters. Definition 2 allows for grouping various objects within different dimensions and time intervals. By generating the set $Clusters = \{C_1, \dots, C_k\}$ of all TimeSC C_i, we a priori permit overlapping clusters. Overlap, however, poses a novel challenge: the set *Clusters* potentially is very large and the contained clusters differ only marginally. In the worst case, two clusters differ only by few objects and hence one of theses clusters provides no novel information. Thus, some clusters may be highly redundant and are not beneficial for the user. As a solution, we aim to extract a subset $Result \subseteq Clusters$ that contains no redundant information.

In a set of clusters $M \subseteq Clusters$ redundancy can be observed, if at least one cluster $C \in M$ exists whose structural properties can be described by the other clusters. More precisely: if we are able to find a set of clusters $M' \subseteq M$ which together group almost the same objects as C and that are located in similar intervals, then C's grouping does not represent novel knowledge.

Definition 3. ***Structural Similarity.*** *A single time series subspace cluster TimeSC $C = (O, \{(d_i, Int_i)_{\{1\dots m\}}\})$ is structural similar to a set of TimeSC M, abbreviated $C \approx M$, iff*

- $\frac{|Obj(M) \cap O|}{|Obj(M) \cup O|} \geq \lambda_{obj}$ *(object coverage)*
- $\forall C_i \in M : \frac{|Int(C_i) \cap Int(C)|}{|Int(C_i) \cup Int(C)|} \geq \lambda_{int}$ *(interval similarity)*

with redundancy parameters $\lambda_{obj}, \lambda_{int} \in [0,1]$, $Obj(M) = \bigcup_{(O_i,\cdot) \in M} O_i$ and $Int(C)$ representing C's intervals via 2-tuples of dimension and point in time: $Int(C) = Int((O,K)) = \{(d,t) \mid \exists (d, Int) \in K : t \in Int\}$.

The higher the redundancy parameter values λ_{obj} and λ_{int} are set, the more time series (λ_{obj}) or intervals (λ_{int}) of C have to be covered by M so that M is considered structural similar to C. In the extreme case of $r_{obj} = r_{dim} = 1$, C's time series and intervals have to be completely covered by M; in this setting, only few clusters are categorized as redundant. By choosing smaller values, redundancy occurs more often.

The final clustering *Result* must not contain structural similar clusters to be redundancy-free. Since, however, several clusterings fulfill this property, we introduce a second structural property that allows us to choose the most-interesting redundancy-free clustering. On the one hand, a cluster is interesting if it contains many objects, i.e. we get a strong generalization. On the other hand, a cluster can represent a long temporal pattern but with less objects, corresponding to a high similarity within the cluster. Since simultaneously maximizing both criteria is contradictory, we introduce a combined objective function that realizes a trade-off between the number of objects and the pattern length:

Definition 4. *The **Interestingness** of a TimeSC $C = (O, \{(d_i, Int_i)_{\{1 \dots m\}}\})$ is defined by*

$$Interest(C) = |O| \cdot \sum_{i=1}^{m} length(Int_i)$$

By adding up the lengths of all intervals, overlap of intervals between different dimensions is rewarded. The optimal clustering result is defined by demanding the two introduced properties:

Definition 5. *The **Optimal Clustering** of the set of all valid clusters Clusters, i.e. $Result \subseteq Clusters$, fulfills*

(1.) redundancy-free property:
$\forall C \in Result : \neg \exists M \subseteq Result : C \approx M \wedge C \notin M$

(2.) maximal interestingness:
For all redundancy-free clusterings $Res' \subseteq Clusters$ it holds
$\sum_{C \in Result} Interest(C) \geq \sum_{C' \in Res'} Interest(C')$.

With this definition of an optimal clustering, the formalization of our novel cluster model for subspace clustering multivariate time series data is complete. In the next section, we will present an efficient algorithm for this model.

4 Efficient Computation

In this section we present an efficient algorithm for the proposed model. Due to space limitations we just present a short overview. Since calculating the optimal clustering according to Def. 5 is NP-hard, our algorithm determines an approximative solution. The general processing scheme is shown in Fig. 3 and basically consists of two cyclically processed phases to determine the clusters. Thus, instead of generating the whole set of clusters *Clusters* and selecting the subset *Result* afterwards, we iteratively generate promising clusters, which are added to the result.

Phase 1: In the first phase of each cycle a set of cluster candidates is generated based on the following procedure: A time series p acting as a prototype for these candidates is randomly selected, and this prototype is contained in each cluster candidate of this cycle. The cluster candidates, i.e. groups of time series O_i, are obtained by successively adding objects x_i to the previous group, i.e. $O_{i+1} = O_i \cup \{x_i\}$ with $O_0 = \{p\}$. Since the interestingness of a cluster depends

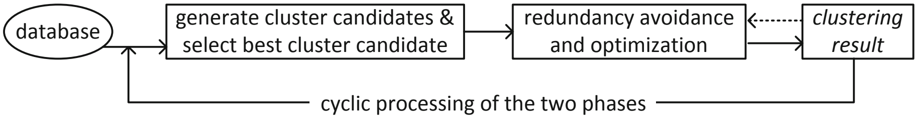

database
generate cluster candidates & select best cluster candidate
redundancy avoidance and optimization
clustering result
cyclic processing of the two phases

Fig. 3. Processing scheme of the algorithm

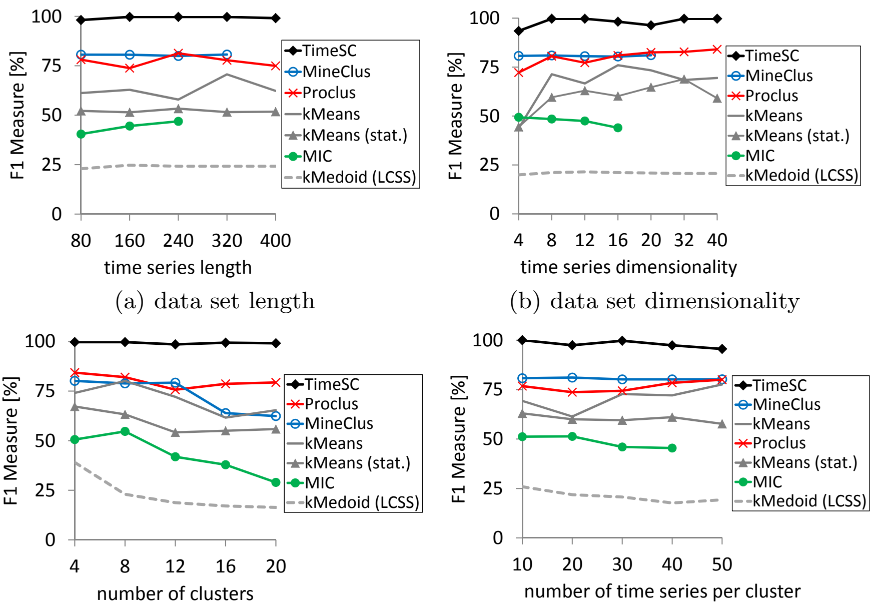
on its size, which is constant for O_i, and the length of the intervals, the choice of x_i is completely determined based on the latter. Accordingly, the best object x_0 is the one which would induce the longest interval interval patterns w.r.t. p (summed over all dimension).

An interval pattern for the prototype p and x_0 at the beginning can potentially include each point in time. Interval patterns for the subsequent objects x_i, however, have to be restricted to the relevant intervals of O_i. Overall, we generate a chain of groups O_i containing objects with a high similarity to p. Based on these candidates we select the set O_+ with the highest interestingness, i.e. according to Def. 4 we combine the size with the interval lengths.

Phase 2: In the second phase, a cluster C for the object set O_+ should be added to the current result Res_j. In the first cycle of the algorithm the result is empty ($Res_0 = \emptyset$), whereas in later cycles it is not. Thus, adding new clusters could induce redundancy. Accordingly, for cluster C we determine those clusters $C' \in Res_j$ with similar relevant intervals (cf. Def. 3). For this set we check if a subset of clusters M covering similar objects as C exists. If not, we can directly add C to the result. In case such a set M exists, we test whether the (summed) interestingness of M is lower than the one of C. In this case, selecting C and removing M is beneficial. As a further optimization we determine the union of C's and M's objects, resulting in a larger cluster U with potentially smaller intervals. If U's interestingness exceeds the previous values, we select this cluster. This procedure is especially useful if clusters of previous iterations are not completely detected, i.e. some objects of the clusters were missed. This step improves the quality of these clusters by adding further objects. Overall, we generate a redundancy-free clustering solution and simultaneously maximize the interestingness as required in Def. 5.

By completing the second phase we initiate the next cycle. In our algorithm the number of cycles is not a priori fixed but it is adapted to the number of detected clusters. We perform $c \cdot |Res_j|$ cycles. The more clusters are detected, the more prototypes should be drawn and the more cycles are performed. Thus, our algorithm automatically adapts to the given data.

In the experimental evaluation, we will demonstrate the efficiency and effectiveness of this algorithm w.r.t. large scale data.



(a) data set length

(b) data set dimensionality

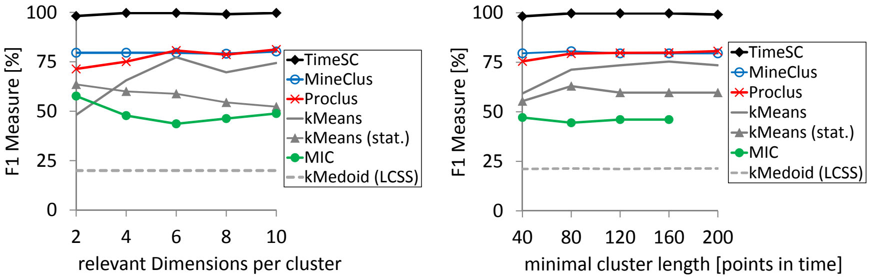
(c) number of clusters

(d) time series per cluster

Fig. 4. Performance under different parameter settings

5 Experiments

We evaluate our TimeSC in comparison to six competing solutions, namely kMeans, a kMeans using statistical features for multivariate time series [19], Proclus [1], MineClus [21], and MIC [16]. We also included kMedoid, where we used the Longest Common Subsequences (LCSS) [18] as a distance measure to allow for partial comparison in the distance computation. We also compared to TriCluster [22], which was provided by the authors on their webpage, but it either delivered no result or the obtained accuracy was very low ($\leq 3\%$); therefore we no longer included it in the experiments. For TimeSC, we used $w = 30$ for the compactness parameter. For the redundancy model, we used $\lambda_{obj} = 0.5$ and $\lambda_{int} = 0.9$. Some of the competing algorithms are not suitable for large datasets; thus, in some experiments, we could not obtain all values. If not stated otherwise, we use the following settings for our synthetic data generator: The dataspace has an extend of [-100,+100], time series length is 200, clusters length is 100, the dataset dimensionality is 10, the number of relevant dimensions per cluster is 5, the number of clusters is 10, the average number of time series per cluster is 25, and there is 10% noise (outliers) in the data. The experiments were performed on AMD Opteron servers with 2.2GHz per core and 256GB RAM. In the experiments, the F1 measure is used to measure accuracy of the obtained clusterings [5,13], and the values are averages of three runs.



(a) relevant dimensions (b) relevant points in time

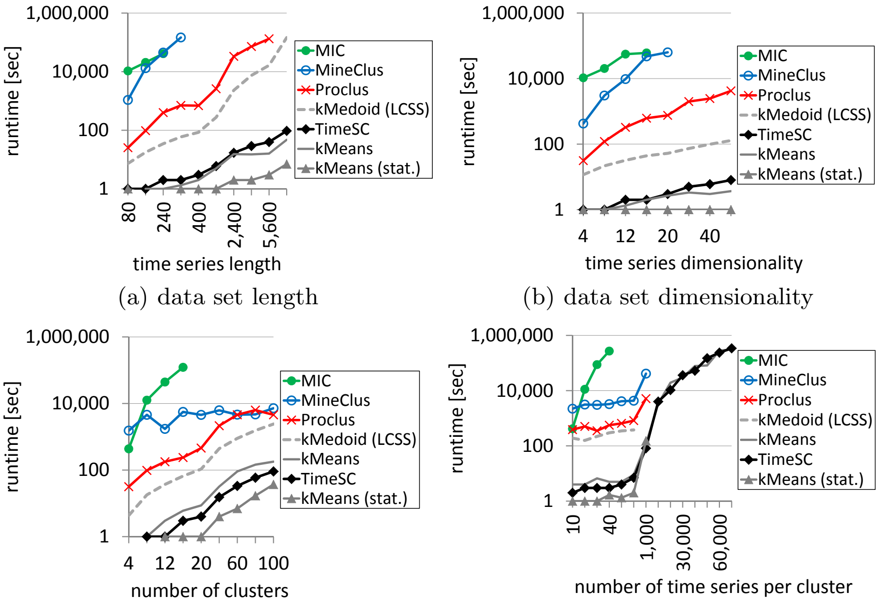
Fig. 5. Performance w.r.t. different number of relevant dimensions and points in time

5.1 Evaluation w.r.t. Effectiveness

Performance on different variations of our standard dataset is analyzed in Fig. 4 and Fig. 5. In Fig. 4(a) and 4(b) we change the dataset size by enlarging either the length of the included time series or the number of attributes per time series. In both experiments, our TimeSC outperforms all competing solutions by a significant margin. Runner ups are the 2D subspace clustering algorithms MineClus and Proclus, which achieve about 80% accuracy. The standard fullspace clustering approach kMeans also performs surprisingly well with about 60% accuracy. The statistical kMeans approach, which was specifically introduced for clustering multivariate time series, however, performs worse than the original kMeans approach. The Triclustering (3D) approach MIC is outperformed by both kMeans variants. This was not expected, as MIC is designed for temporal data. And finally, the LCSS-based kMedoid only achieves about 20% accuracy in both experiments.

Next, we enlarge the data by increasing the number of clusters (Fig. 4(c)) and by increasing the number of time series per cluster (Fig. 4(d)). With an increasing number of clusters, TimeSC and Proclus achieve stable results, while the accuracies of the other competing approaches continuously sink. For an increasing number of time series per cluster, all the algorithms achieve stable results. Overall, TimeSC outperforms the competing solutions in all settings.

In Fig. 5(a) and 5(b) we change the number of relevant dimensions per cluster and the number of relevant points in time per cluster, i.e. the minimal cluster length. Overall, the obtained accuracies are similar to the preceding experiments. TimeSC outperforms the other methods, and from these methods only the 2D subspace clustering algorithms can achieve stable results of about 80% accuracy. Also, as expected, with increasing relevant dimensions and relevant points in time, finding clusters in the data becomes simpler which is expressed by the strong increase in accuracy for the standard kMeans algorithm. This, however, does not hold for the statistical kMeans, whose accuracy sinks with increasing relevant dimensions and points in time.



(a) data set length

(b) data set dimensionality

(c) number of clusters

(d) number of time series per cluster

Fig. 6. Efficiency comparison (log. scale)

5.2 Evaluation w.r.t. Efficiency

Our algorithm is designed for larger datasets. To show this, we scaled the experiments from Fig. 4 to much higher values, as shown in Fig. 6. For example, the database size for the last step (70, 000 time series) in Fig. 6(d) is 12.32 GB. The experiments illustrate that only the kMeans algorithms and TimeSC are suitable for the larger datasets. Due to the high runtimes, many values could not be obtained for the other approaches. From the subspace and triclustering algorithms, only Proclus shows acceptable runtimes, while the results of MineClus and MIC indicate that these algorithms are not applicable for larger datasets.

6 Conclusion

We introduced a novel model for subspace clustering of multivariate time series data. The clusters in our model are formed by individual sets of relevant intervals per dimension, which together fulfill temporal coherence. We develop a redundancy model to avoid structurally similar clusters and introduce an approximate algorithm for generating clusterings according to our novel model. In the experimental comparison, we showed that our approach is efficient and generates clusterings of higher quality than the competing methods.

Acknowledgments. This work has been supported by the UMIC Research Centre, RWTH Aachen University.

References

1. Aggarwal, C.C., Procopiuc, C.M., Wolf, J.L., Yu, P.S., Park, J.S.: Fast algorithms for projected clustering. In: ACM SIGMOD, pp. 61–72 (1999)
2. Dasu, T., Swayne, D.F., Poole, D.: Grouping Multivariate Time Series: A Case Study. In: IEEE ICDMW, pp. 25–32 (2005)
3. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. PVLDB 1(2), 1542–1552 (2008)
4. Fu, T.: A review on time series data mining. Engineering Applications of Artificial Intelligence 24(1), 164–181 (2011)
5. Günnemann, S., Färber, I., Müller, E., Assent, I., Seidl, T.: External evaluation measures for subspace clustering. In: ACM CIKM, pp. 1363–1372 (2011)
6. Hu, Z., Bhatnagar, R.: Algorithm for discovering low-variance 3-clusters from real-valued datasets. In: IEEE ICDM, pp. 236–245 (2010)
7. Jiang, D., Pei, J., Ramanathan, M., Tang, C., Zhang, A.: Mining coherent gene clusters from gene-sample-time microarray data. In: ACM SIGKDD, pp. 430–439 (2004)
8. Jiang, H., Zhou, S., Guan, J., Zheng, Y.: gTRICLUSTER: A More General and Effective 3D Clustering Algorithm for Gene-Sample-Time Microarray Data. In: Li, J., Yang, Q., Tan, A.-H. (eds.) BioDM 2006. LNCS (LNBI), vol. 3916, pp. 48–59. Springer, Heidelberg (2006)
9. Kremer, H., Kranen, P., Jansen, T., Seidl, T., Bifet, A., Holmes, G., Pfahringer, B.: An effective evaluation measure for clustering on evolving data streams. In: ACM SIGKDD, pp. 868–876 (2011)
10. Kriegel, H. P., Kröger, P., Zimek, A.: Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM TKDD 3(1) (2009)
11. Liao, T.W.: Clustering of time series data - a survey. Pattern Recognition 38(11), 1857–1874 (2005)
12. Minnen, D., Starner, T., Essa, I.A., Isbell, C.: Discovering characteristic actions from on-body sensor data. In: IEEE ISWC, pp. 11–18 (2006)
13. Müller, E., Günnemann, S., Assent, I., Seidl, T.: Evaluating clustering in subspace projections of high dimensional data. PVLDB 2(1), 1270–1281 (2009)
14. Oates, T.: Identifying distinctive subsequences in multivariate time series by clustering. In: ACM SIGKDD, pp. 322–326 (1999)
15. Procopiuc, C.M., Jones, M., Agarwal, P.K., Murali, T.M.: A monte carlo algorithm for fast projective clustering. In: ACM SIGMOD, pp. 418–427 (2002)
16. Sim, K., Aung, Z., Gopalkrishnan, V.: Discovering correlated subspace clusters in 3D continuous-valued data. In: IEEE ICDM, pp. 471–480 (2010)
17. Singhal, A., Seborg, D.: Clustering multivariate time-series data. Journal of Chemometrics 19(8), 427–438 (2005)
18. Vlachos, M., Gunopulos, D., Kollios, G.: Discovering similar multidimensional trajectories. In: IEEE ICDE, pp. 673–684 (2002)
19. Wang, X., Wirth, A., Wang, L.: Structure-based statistical features and multivariate time series clustering. In: IEEE ICDM, pp. 351–360 (2007)
20. Wu, E.H.C., Yu, P.L.H.: Independent Component Analysis for Clustering Multivariate Time Series Data. In: Li, X., Wang, S., Dong, Z.Y. (eds.) ADMA 2005. LNCS (LNAI), vol. 3584, pp. 474–482. Springer, Heidelberg (2005)
21. Yiu, M.L., Mamoulis, N.: Frequent-pattern based iterative projected clustering. In: IEEE ICDM, pp. 689–692 (2003)
22. Zhao, L., Zaki, M.J.: TriCluster: An effective algorithm for mining coherent clusters in 3D microarray data. In: ACM SIGMOD, pp. 694–705 (2005)

A Vertex Similarity Probability Model for Finding Network Community Structure

Kan Li[1] and Yin Pang[1,2]

[1] Beijing Key Lab of Intelligent Information Technology, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, 100081, China
likan@bit.edu.cn
[2] Beijing Institute of Tracking and Telecommunication Technology, Beijing, 100094, China
pangyin@bit.edu.cn

Abstract. Most methods for finding community structure are based on the prior knowledge of network structure type. These methods grouped the communities only when known network is unipartite or bipartite. This paper presents a vertex similarity probability (VSP) model which can find community structure without priori knowledge of network structure type. Vertex similarity, which assumes that, for any type of network structures, vertices in the same community have similar properties. In the VSP model, "Common neighbor index" is used to measure the vertex similarity probability, as it has been proved to be an effective index for vertex similarity. We apply the algorithm to real-world network data. The results show that the VSP model is uniform for both unipartite networks and bipartite networks, and it is able to find the community structure successfully without the use of the network structure type.

Keywords: community structure, type of the network structure, vertex similarity, common neighbor index.

1 Introduction

As part of the recent surge of research on large, complex networks, attention has been devoted to the computational analysis of complex networks [1-4]. Complex networks, such as social networks and biological networks, are all highly dynamic objects which grow and change quickly over time. These networks have a common feature, namely "community structure". Communities, also known as clusters or modules, are groups of vertices which could share common properties and/or have similar roles within the graph [5]. Finding community structure and clustering vertices in the complex network, is key to learning a complex network topology, to understanding complex network functions, to founding hidden mode, to link prediction, and to evolution detection. Through the analysis of community structure, researchers have achieved a lot results, such as in [6, 7], V. Spirin *et al.* revealed the relationship between protein function and interactions inherent; in [8, 9], Flake *et al.* found the internal relations of hyperlink and the main page; in [10, 11], Moody *et al.* identified the social organizations to evolve over time and so on.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 456–467, 2012.

The most popular method for finding community structure is the modularity matrix method [12, 13] proposed by Newman *et al.* which is based on spectral clustering. The Modularity model proves that, if the type of the network structure is known, modularity optimization is able to find community structure in both unipartite and bipartite networks by the maximum or minimum eigenvalue separately. Then, some scientists have sought to detect the community in bipartite networks like Michael J. Barber [14]. BRIM proposed by Barber and his colleagues can determine the number of communities of a bipartite network. Furthermore, in [15], Barber and Clark use the label-propagation algorithm (LPA) for identifying network communities. However, [14, 15] can not be used without knowing the type of network.

There are other methods to find community structure. Hierarchical clustering is adopted frequently in finding community structures, in which vertices are grouped into communities that further are subdivided into smaller communities, and so forth, as in [12]. Clauset, Moore and Newman propose HRG [16] using the maximum likelihood estimation to forecast the probability of connections between vertices. Hierarchical methods perform remarkably in clear hierarchy network, but not so impressive under contrary circumstance. Moreover, a hierarchical method always has high computational complexity. In 2009, Roger Guimera and Marta Sales-Pardo proposed a stochastic block model [17] based on HRG. Different from traditional concept which divide network by principle of "inside connection dense outside sparse", in [17], the probability that two vertices are connected depends on the blocks to which they belong. However, the assumption that vertices in same blocks have same connection probability is not accurate. Recently, Karrer and Newman [18] also proposed a stochastic block model which considers the variation in vertex degree. This stochastic block model solves the heterogeneous vertex degrees problem and got a better result than other previous researches without degree correction. It can be used in both types of networks, but different types of networks should be dealt with separately none the less.

In some cases, researchers have no priori knowledge of the network structure. For example, when we know the interaction of vertex in the protein network, we may have no knowledge of the network structure type. Moreover, when we get a network which consists of people's relationships in schools, the type of network may not be sure. It is because that if links only exist between students, the network will be a unipartite network; or if links exist between students and teachers, the network will be a bipartite one. An effective method used for finding community structure in both unipartite and bipartite networks is needed.

It is discussed before that most methods deal with the unipartite network or bipartite network separately, because the properties of networks are different in different types of the network structure. Unipartite networks assume that connections between the vertices in same community are dense, and between the communities are sparse, such as Social network [19], biochemical network [20] and information network [21]. However, some real networks are bipartite with edges joining only vertices of different communities, such as shopping networks [22], protein-protein interaction networks [23], plant-animal mutualistic networks [24], scientific publication networks [25], etc. Although the properties of "edges" in the two types of networks are different, vertices in the same communities should be similar because vertices in same communities have similar properties. In this paper, we develop a uniform VSP model which is based on the vertex similarity. Therefore, the VSP model can be used in any type of networks as long as we put similar vertices in same communities. The VSP model gets ideal result both by theoretical proof and experimental analysis.

The paper is organized as follows. In section 2, we prove vertex similarity theory is suitable for finding community structure. We present the VSP model and the method to group network into two communities in section 3. In section 4, we make the experiment in both unipartite and bipartite network. Compared with Newman's modularity, the VSP model is an accurate uniform model which can find community structure without prior knowledge of type of the network structure. Finally, we draw our conclusions.

2 Vertex Similarity in Finding Community Structure

The concept of community informs that vertices in the same community should share common properties no matter in unipartite or bipartite network. It means that vertices in the same community should be similar, although edges in different type of the network structures are connected in different ways. Therefore, we change our focus from "edges" to "vertices" for finding communities.

Vertex similarity is widely studied by researchers in complex network. It is sometimes called structural similarity, to distinguish it from social similarity, textual similarity, or other similarity types. It is a basic premise of research on networks that the structure of a network reflects real information about the vertices the network connects, so it is reasonable that meaningful structural similarity measures might exist [26]. In general, if two vertices have a number of common neighbors, we believe that these two vertices are similar. In community detection, we assume that two similar vertices have similar properties and should be grouped in the same community.

Let Γ_x be the neighborhood of vertex *x* in a network, i.e., the set of vertices that are directly connected to *x* via an edge. Then $|\Gamma_x \cap \Gamma_y|$ is the number of common neighbors of *x* and *y*. Common neighbor index, Salton index, Jaccard index, Sorenson index, LHN (Leicht-Holme-Newman) index, and Adamic-Adar index [27-31] are five famous methods for vertex similarity. Many researchers have analyzed and compared these methods. Liben-Nowell[32] and Zhou Tao[33] proved that the simplest measurement "common neighbor index" performs surprisingly well. We use "common neighbor index" to measure the vertex similarity in our VSP model.

Definition 1. For two vertices *x* and *y*, if there is a vertex *z* to be the neighbor of *x* and *y* at the same time, we call *x* and *y* a pair, denoted as pair(*x, y*). *z* is called the common neighbor of pair(*x, y*).

Since vertices which are in the same community have similar properties, we assume vertices in the same community are similar vertices. The more similar the vertices inside a community are the more common neighbors they have. The number of common neighbors N_{ij} of vertices *i* and *j* is given by,

$$N_{ij} = |\Gamma_i \cap \Gamma_j|, \text{ and } N_{ii} = 0 .$$

The sum of common neighbors with vertices in same communities N_{in} is given by

$$N_{in} = \sum_{\substack{i,j \in \text{same} \\ \text{commumity}}} N_{ij} \cdot$$

And the sum of common neighbors with vertices in different communities N_{out} is given by

$$N_{out} = \sum_{\substack{i,j \notin \text{same} \\ \text{commumity}}} N_{ij} \cdot$$

Therefore, the task of maximizing the number of common neighbors in the same community is to get $\max(N_{in})$ or to get $\min(N_{out})$. The sum of common neighbors in the network *R* is given by

$$R = \frac{1}{2}\sum_{i,j \in n} N_{ij} \cdot$$

We define the adjacency matrix *A* to be the symmetric matrix with elements A_{ij}. If there is an edge joining vertices *i* and *j*, $A_{ij} = 1$; if no, $A_{ij} = 0$. Define $\mathbf{a_i}$ as *i* th vector of *A*, so as *A* can be rewritten as $A = [\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_n}]$. If and only if $A_{ik}A_{kj} = 1$, the vertex k is a common neighbor of vertices *i* and *j*. Therefore N_{ij} can be rewritten as

$$N_{ij} = \sum_k A_{ik}A_{kj} = \mathbf{a_i} \cdot \mathbf{a_j},$$

when *i* and *j* are two different vertices. As $\mathbf{a_i} \cdot \mathbf{a_i} = k_i$, matrix *N* is

$$N = A^T A - \Lambda_k,$$

where $\Lambda_k = diag(k_1, k_2, ..., k_n)$. It allows us to rewrite *R* as

$$\begin{aligned} R &= \frac{1}{2}\sum_{\substack{i,j \in n \\ i \neq j}} \mathbf{a_i} \cdot \mathbf{a_j} \\ &= \frac{1}{2}(\sum_{i,j \in n} \mathbf{a_i} \cdot \mathbf{a_j} - \sum_i k_i) \\ &= \frac{1}{2}(\sum_i \mathbf{a_i} \cdot (k_1, k_2, ..., k_n)^T - \sum_i k_i) \\ &= \frac{1}{2}((k_1, k_2, ..., k_n) \cdot (k_1, k_2, ..., k_n)^T - \sum_i k_i) \\ &= \sum_i \frac{1}{2} k_i (k_i - 1) \end{aligned} \quad (1)$$

Definition 2. According to Eq.(1), *R* is only related to a function of vertex degree. To analyze the relationship between a vertex x and common neighbor index, we define the function as a common neighbor degree index, denoted as c_x. Let $c_x = k_x(k_x - 1)/2$. Therefore, $R = \sum_{x \in n} c_x$.

Total number of common neighbors in the network equals the number of common neighbors in same communities plus the number of common neighbors different communities, *R* also can be written as $R = N_{in} + N_{out}$.

The following proves that using common neighbor index in finding community structure is suitable in both unipartite networks and bipartite networks.

2.1 Common Neighbor Index in Unipartite Network

For a unipartite network, the basic community detection principle is "edges inside communities are dense, outside are sparse". Let the sum of edges with vertices in different communities is A_{out}, where

$$A_{out} = \sum_{\substack{i,j \notin \text{same} \\ \text{commumity}}} A_{ij} \cdot$$

The task is to minimize A_{out}, written as $\min(A_{out})$.

Suppose *i* and *j* are two vertices in different communities. If *i* and *j* are connected, there are $k_i - 1$ pairs (where k_i is the degree of *i*) with a common neighbor *i*, each of which is formed by *j* and a neighbor of *i*. In a unipartite network, neighbors of a vertex are almost in the same community. As a result, for *i*, most of its neighbor should be in the same community with *i* except *j* (if *j* is a neighbor of *i*). As shown in Fig. 1.

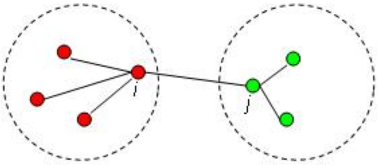


Fig. 1. An example of two vertices in different communities in a unipartite network

If *i* and *j* are not connected, no common neighbor is counted. Therefore the number of common neighbors with pairs of vertices in different communities is

$$N_{out} = \sum_{\substack{i,j \notin \text{same} \\ \text{commumity}}} A_{ij}(k_i - 1) \tag{2}$$

A is symmetric which allows us to rewrite Eq.(2) as

$$N_{out} = \sum_{\substack{i,j \notin \text{same} \\ \text{commumity} \\ \text{and i<j}}} A_{ij}(k_i + k_j - 2) \tag{3}$$

For two vertices *i* and *j* in different communities, N_{out} is related to A_{ij}, k_i and k_j. If there is an edge between *i* and *j*, A_{out} will plus 1 and N_{out} will plus $k_i + k_j - 2$. As $k_i + k_j - 2 \geq 0$, we consider A_{out} and N_{out} have the same growth trend. It means getting $\min(N_{out})$ is equivalent to getting $\min(A_{out})$. The conclusion is in line with the basic principles of the unipartite network community detection.

2.2 Common Neighbor Index in Bipartite Network

For a bipartite network, the basic community detection principle is “edges inside communities are sparse, outside are dense”. The task is to maximize A_{out}, written as $\max(A_{out})\cdot$

In a bipartite network, almost all adjacent vertices are in different communities. For a pair of vertices which are in the same community, the common neighbor should be in a different community, as shown in Fig. 2.

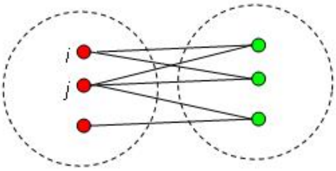


Fig. 2. An example of two vertices in same communities in a bipartite network

As a result, for any pair of vertices *i* and *j* which are in the same community, N_{ij} have $2N_{ij}$ edges between different communities. In the overall network, each edge will be counted $(k_i + k_j - 2)$ times.

$$2N_{in} = \sum\nolimits_{\substack{i,j \notin \text{same} \\ \text{commumity}}} A_{ij}(k_i + k_j - 2)\cdot \tag{4}$$

A is symmetric which allows us to rewrite Eq.(4) as

$$N_{in} = \sum\nolimits_{\substack{i,j \notin \text{same} \\ \text{commumity} \\ \text{and i<j}}} A_{ij}(k_i + k_j - 2) \tag{5}$$

Similar as section 2.1, we consider A_{out} and N_{in} have same growth trend. It means getting $\max(N_{in})$ is equivalent to getting $\max(A_{out})$. The conclusion is in line with the basic principles of the bipartite network community detection.

In summary of section 2.1 and 2.2, the common neighbor index of vertex similarity is suitable for finding community structure in both unipartite and bipartite networks.

3 A VSP Model for Finding Community Structure

In this section, we propose our VSP model to find community structure. In [13], Newman *et al.* proved that a good division of a network in to communities “in which the number of edges inside groups is bigger than expected”. It can get a better result than the measures based on pure numbers of edges between communities. Similarly, a good division of a network into communities should be one which the number of common neighbors within communities is bigger than expected. Let

$$Q=\begin{matrix}\text{(common neighbors within communities-}\\ \text{expected number of such common neighbors)}\end{matrix}. \tag{6}$$

It is a function that divides the network into groups, with larger values indicating stronger community structure. We build a random network in which vertices have same common neighbor degrees as the vertices in the complex network, and assume the expected number of common neighbors as the number in the random network. In section 2, we have proved that common neighbor index can be used to find communities instead of edges. However, we can also find that N_{out} in unipartite network and N_{in} in bipartite network are both affected not only by edges but also by vertex degree. It is known that common neighbor degree c_i is a function of k_i and R is the sum of c_i. We use c_i to calculate the common neighbors in the random network. The probability of a random vertex to be a common neighbor of a particular vertex i depends only on the expected common neighbor degree c_i. The probabilities of a random vertex to be a common neighbor of two vertices are independent on each other. This implies that the expected number of common neighbors P_{ij} between vertices i and j is the product $f(c_i)f(c_j)$ of separate functions of the two common neighbor degrees, where the functions must be the same since P_{ij} is symmetric. Hence $f(c_i)=Cc_i$ for some constant C,

$$\sum\nolimits_{i,j\in n} P_{ij} = \sum\nolimits_i f(c_i) \sum\nolimits_j f(c_j) = C^2 R^2 \tag{7}$$

Vertices in random network have the same common neighbor degree just like in complex network, $\sum\nolimits_{i,j\in n} P_{ij} = \sum\nolimits_{i,j\in n} N_{ij} = 2R$. So, $C=\sqrt{\frac{2}{R}}$ and

$$f(c_i)=\sqrt{\frac{2}{R}}c_i \tag{8}$$

We get the expected number of common neighbors of pair(*x, y*) as follows,

$$P_{ij} = f(c_i)f(c_j) = \frac{2c_x c_y}{R}. \tag{9}$$

The VSP model can be written,

$$Q=\frac{1}{2R}\sum\nolimits_{\substack{i,j\in\text{same}\\ \text{community}}} [N_{ij} - \frac{2c_i c_j}{R}]. \tag{10}$$

What we should notice is that,

$$\sum\nolimits_{i,j\in n} \frac{2c_i c_j}{R} = 2\frac{\sum\nolimits_{j\in n} c_j \sum\nolimits_{i\in n} c_i}{R} = 2R = \sum\nolimits_{i,j\in n} N_{ij}. \tag{11}$$

Thus,

$$\frac{1}{2}\sum_{i,j\in n}[N_{ij}-\frac{2c_i c_j}{R}]=0 \tag{12}$$

Let

$$B_{ij}=N_{ij}-\frac{2c_i c_j}{R}. \tag{13}$$

B is the VSP matrix, and $\sum_{i,j\in n}B_{ij}=0$.

We use the VSP matrix instead of modularity matrix to find the community structure. In the VSP model, the higher value of Q, the more similar vertices are in the same community. It can be applied to both unipartite networks and bipartite networks without knowing the exact type of network structure in advance. It is more flexible than the previous methods which deal with the grouping separately according to the type of the network structure.

4 Experimental Results

In this section, we apply the VSP model to a unipartite network and two bipartite networks with Pajek [34]. The unipartite network shows the dolphin social network studied by Lusseau *et al.* [35]. The bipartite networks show the interactions of women in the American Deep South at various social events [36] and Scotland Corporate Interlock in early twentieth century [37].

Since we know the actual communities for the real networks, we measure the accuracy of the VSP model by directly comparing with the known communities. We take use of the normalized mutual information I_{normn} [38] for the comparison. When the found communities match the real ones, we have I_{norm}=1, and when they are independent of the real ones, we have I_{norm}=0.

We compare the VSP model with the Modularity model in unipartite networks and bipartite networks by three properties: the edges outside communities; Q of the Modularity model, where Q is the edges within communities minus expected number of such edges, written as *Q-Modularity*; and I_{norm}.

4.1 Finding Community Structure in Unipartite Network

Q in Eq. (6) is written as *Q-VSP* in this section. For a unipartite network, the VSP model maximizes *Q-VSP* to find the community structure, while the Modularity model maximizes *Q-Modularity*.

The dolphin social network is a classical unipartite social network. The vertices in this network represent 62 bottlenose dolphins living in Doubtful Sound, New Zealand, with social ties between dolphin pairs established by direct observation over a period of several years. It is used a lot in community detection because the dolphin group split into two smaller subgroups following the departure of the population. Fig.3 shows the clustering results using the VSP model and the Modularity model respectively.

Two red vertices are grouped into the green community in the VSP model, while three red vertices are grouped into the green community in the Modularity model.

In a unipartite network, edges inside the communities are dense, while outside are sparse. Edges outside communities should be small; *Q-Modularity* should be large; I_{norm} close to 1. Properties of the dolphin social network are shown in Table.1. It shows that two properties of VSP model are better than the Modularity model in this unipartite network. *Q-Modularity* of the VSP model is 0.381 which is approximately equal to the one of the Modularity model. The VSP model performs well in unipartite networks.

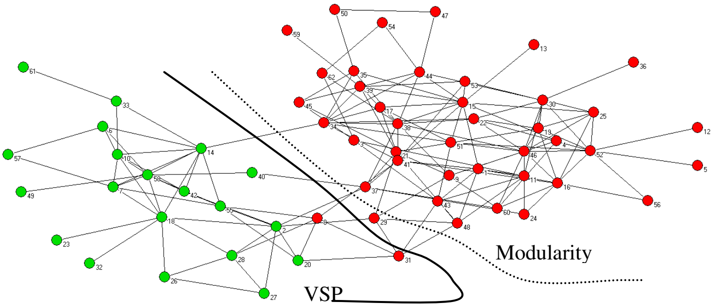


Fig. 3. Finding community structures of the dolphin social network. The red and green vertices represent the division of the network. The solid curve represents the division of the VSP model.. The dotted curve represents the division of the Modularity model.

Table 1. Properties of the dolphins social network

	Edges outside communities	Q-Modularity	I_{norm}
VSP	8	0.381	0.813
Modularity	9	0.386	0.752

4.2 Finding Community Structure in Bipartite Network

In a bipartite network, the VSP model also maximizes *Q-VSP* to find the community structure, the Modularity model minimizes *Q-Modularity*, which is contrary to in the unipartite network.

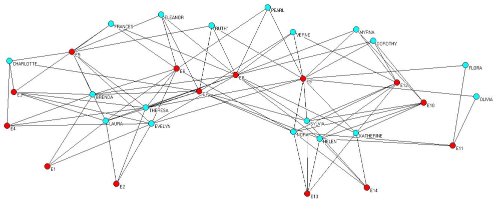


Fig. 4. Find community structures of Southern women network using the VSP model

The Southern women data set describes the grouping of 18 women in 14 social events constitute a bipartite network; and an edge exists between a woman and a social event if the woman was in attendance at the event. We use this network here to group it into two communities, shown in Fig.4. It shows that the VSP model groups the network accurately into two communities of "women" and "events".

Although using other finding community structure methods can also get the same result, they should know the type of the network in advance. For example, in modularity, it gets the smallest value of the Modularity model but not the biggest because the southern women network is a bipartite network.

As a second example of bipartite network, we consider a data set on Scotland Corporate Interlock in early twentieth century. The data set is characterized by 108 Scottish firms, detailing the corporate sector, capital, and board of directors for each firm. The data set includes only those board members who held multiple directorships, totaling 136 individuals. Unlike the Southern women network, the Scotland corporate interlock is not connected. We got the division of one community with 102 vertices and the other with 142 vertices when *Q-VSP* is the maximum value.

We also compare three properties of the VSP model and the Modularity model. In a bipartite network, edges inside the communities are sparse, while outside are dense. Edges outside communities should be large; *Q-Modularity* should be small; I_{norm} close to 1. Properties of the Scotland corporate interlock are shown in Table 2. All the three properties of the VSP model are better than the Modularity model. It proves that the VSP model also finds community structure accurately in bipartite networks.

Table 2. Properties of Scotland corporate interlock network

	Edges inside communities	Q-Modularity	I_{norm}
VSP	47	-0.372	0.767
Modularity	150	-0.169	0.377

In summary of section 4.1 and 4.2, the VSP model is a uniform model for finding community structure which can be used in both unipartite networks and bipartite networks. It is flexible and applicable to a wide range. For instance, in the protein network which people only knows its tip of iceberg, the VSP model can find the community structure only with the topology of the network, even when we have no idea of the type of the network structure.

5 Conclusion

In this paper, we define a VSP model for finding the community structure in complex networks. The VSP model is based on the vertex similarity using the common neighbor index. As common neighbor index is proved an effective measurement of the vertex similarity methods in complex network, it is applied to the VSP model to measure the vertex similarity. We prove that calculating the common neighbor inside communities of the network is equivalent to calculation the least edges outside communities in a unipartite network and the most edges outside communities in a bipartite network.

Therefore, it is suitable for finding community structure in both unipartite and bipartite network. Then we give the expectation of the common neighbor between any two vertices and gave the VSP model. At last, we apply our model in the dolphin social network, Southern women event network and Scotland corporate interlock network separately. Results showed that the VSP model is effective for finding community structure without the need of the network structure type.

Acknowledgments. The research was supported by Natural Science Foundation of China (No.60903071).

References

1. Wasserman, S., Faust, K.: Social Networks Analysis. Cambridge University Pres, Cambridge (1994)
2. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.U.: Complex networks: Structure and dynamics. Physics Reports-Review Section of Physics Letters 424, 175–308 (2006)
3. Lu, L.Y., Zhou, T.: Link prediction in complex networks: A survey. Physica a-Statistical Mechanics and Its Applications 390, 1150–1170 (2011)
4. Boguna, M., Krioukov, D., Claffy, K.C.: Navigability of complex networks. Nature Physics 5, 74–80 (2009)
5. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17, 734–749 (2005)
6. Spirin, V., Mirny, L.A.: Protein complexes and functional modules in molecular networks. Proceedings of the National Academy of Sciences of the United States of America 100, 12123–12128 (2003)
7. Chen, J.C., Yuan, B.: Detecting functional modules in the yeast protein-protein interaction network. Bioinformatics 22, 2283–2290 (2006)
8. Flake, G.W., Lawrence, S., Giles, C.L., Coetzee, F.M.: Self-organization and identification of web communities. Computer 35, 66–71 (2002)
9. Dourisboure, Y., Geraci, F., Pellegrini, M.: Extraction and classification of dense communities in the web. In: Proceedings of the 16th International Conference on the World Wide Web, pp. 461–470. ACM, New York (2007)
10. Moody, J., White, D.R.: Structural cohesion and embeddedness: A hierarchical concept of social groups. American Sociological Review 68, 103–127 (2003)
11. Wellman, B.: The development of social network analysis: A study in the sociology of science. Contemporary Sociology-a Journal of Reviews 37, 221–222 (2008)
12. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E 69 (2004)
13. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Physical Review E 74 (2006)
14. Barber, M.J.: Modularity and community detection in bipartite networks. Physical Review E 76 (2007)
15. Barber, M.J., Clark, J.W.: Detecting network communities by propagating labels under constraints. Physical Review E 80 (2009)
16. Clauset, A., Moore, C., Newman, M.E.J.: Hierarchical structure and the prediction of missing links in networks. Nature 453, 98–101 (2008)

17. Guimera, R., Sales-Pardo, M.: Missing and spurious interactions and the reconstruction of complex networks. Proceedings of the National Academy of Sciences of the United States of America 106, 22073–22078 (2009)
18. Karrer, B., Newman, M.E.J.: Stochastic blockmodels and community structure in networks. Physical Review E 83 (2011)
19. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America 99, 7821–7826 (2002)
20. Holme, P., Huss, M., Jeong, H.W.: Subnetwork hierarchies of biochemical pathways. Bioinformatics 19, 532–538 (2003)
21. Rosvall, M., Bergstrom, C.T.: An information-theoretic framework for resolving community structure in complex networks. Proceedings of the National Academy of Sciences of the United States of America 104, 7327–7331 (2007)
22. Chuma, J., Molyneux, C.: Coping with the costs of illness: The role of shops and shopkeepers as social networks in a low-income community in coastal kenya. Journal of International Development 21, 252–270 (2009)
23. Li, F., Long, T., Lu, Y., Quyang, Q., Tang, C.: The yeast cell-cycle network is robustly designed. PNAS 101(14), 4781–4786 (2004)
24. Bascompte, J., Jordano, P., Melian, C.J., Olesen, J.M.: The nested assembly of plant-animal mutualistic networks. Proceedings of the National Academy of Sciences of the United States of America 100, 9383–9387 (2003)
25. Guimera, R., Uzzi, B., Spiro, J., Amaral, L.A.N.: Team assembly mechanisms determine collaboration network structure and team performance. Science 308, 697–702 (2005)
26. Leicht, E.A., Holme, P., Newman, M.E.J.: Vertex similarity in networks. Physical Review E 73 (2006)
27. Salton, G., McGill, M.J.: Introduction to modern information retrieval. MuGraw-Hill, Auckland (1983)
28. Jaccard, P.: Nouvelles recherches sur la distribution florale. Bulletin de la Societe Vaudoise des Science Naturelles 44, 223–270 (1908)
29. Sφrensen, T.: A method of establishing groups of equal amplitude in plant sociology based on similarity of species content, and its application to analyses of the vegetation on Danish commons. Det Kongelige Danske Videnskabernes Selskab. Biologiske Skrifter 5(4), 1–34 (1948)
30. Salton, G.: Automatic text processing: The transformation, analysis, and retrieval of information by computer. Addison-Wesley, Boston (1989)
31. Adamic, L.A., Adar, E.: Friends and neighbors on the Web. Social Networks 25, 211–230 (2003)
32. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. Journal of the American Society for Information Science and Technology 58, 1019–1031 (2007)
33. Zhou, T., Lu, L.Y., Zhang, Y.C.: Predicting missing links via local information. European Physical Journal B 71, 623–630 (2009)
34. Pajek: `http://vlado.fmf.uni-lj.si/pub/networks/pajek`
35. Lusseau, D., Schneider, K., Boisseau, O.J., Haase, P., Slooten, E., Dawson, S.M.: The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations - Can geographic isolation explain this unique trait? Behavioral Ecology and Sociobiology 54, 396–405 (2003)
36. Davis, A., Gardner, B.B., Gardner, M.R.: Deep South. University of Chicago Press (1941)
37. Scott, J., Hughes, M.: The anatomy of Scottish capital: Scottish companies and Scottish capital. Croom Helm, London (1980)
38. Danon, L., Diaz-Guilera, A., Arenas, A.: The effect of size heterogeneity on community identification in complex networks. Journal of Statistical Mechanics-Theory and Experiment, P11010 (2006)

Hybrid-ε-greedy for Mobile Context-Aware Recommender System

Djallel Bouneffouf, Amel Bouzeghoub, and Alda Lopes Gançarski

Department of Computer Science, Télécom SudParis,
UMR CNRS Samovar, 91011 Evry Cedex, France
{Djallel.Bouneffouf,Amel.Bouzeghoub,
Alda.Gancarski}@it-sudparis.eu

Abstract. The wide development of mobile applications provides a considerable amount of data of all types. In this sense, Mobile Context-aware Recommender Systems (MCRS) suggest the user suitable information depending on her/his situation and interests. Our work consists in applying machine learning techniques and reasoning process in order to adapt dynamically the MCRS to the evolution of the user's interest. To achieve this goal, we propose to combine bandit algorithm and case-based reasoning in order to define a contextual recommendation process based on different context dimensions (social, temporal and location). This paper describes our ongoing work on the implementation of a MCRS based on a hybrid-ε-greedy algorithm. It also presents preliminary results by comparing the hybrid-ε-greedy and the standard ε-greedy algorithm.

Keywords: Machine learning, contextual bandit, personalization, recommender systems, exploration/exploitation dilemma.

1 Introduction

Mobile technologies have made access to a huge collection of information, anywhere and anytime. Thereby, information is customized according to users' needs and preferences. This brings big challenges for the Recommender System field. Indeed, technical features of mobile devices yield to navigation practices which are more difficult than the traditional navigation task.

A considerable amount of research has been done in recommending relevant information for mobile users. Earlier techniques [8, 10] are based solely on the computational behavior of the user to model his interests regardless of his surrounding environment (location, time, near people). The main limitation of such approaches is that they do not take into account the dynamicity of the user's context. This gives rise to another category of recommendation techniques that tackle this limitation by building situation-aware user profiles. However, these techniques have some problems, namely how to recommend information to the user in order to follow the evolution of his interest.

In order to give Mobile Context-aware Recommender Systems (MCRS) the capability to provide the mobile user information matching his/her situation and adapted to the evolution of his/her interests, our contribution consists of mixing bandit algorithm (BA) and case-based reasoning (CBR) methods in order to tackle these two issues:

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 468–479, 2012.

- Finding situations that are similar to the current one (CBR);
- Making the deal between exploring the user interests and recommending the most relevant content according to the current situation (BA).

The remainder of the paper is organized as follows. Section 2 reviews some related works. Section 3 presents the proposed recommendation algorithm. The experimental evaluation is described in Section 4. The last Section concludes the paper and points out possible directions for future work.

2 Background

We reference in the following recent relevant recommendation techniques that tackle the both issues namely: following the evolution of user's interests and managing the user's situation.

2.1 Following the Evolution of User's Interests

The trend today on recommender systems is to suggest relevant information to users, using supervised machine learning techniques. In these approaches, the recommender system has to execute two steps: (1) The learning step, where the system learns from samples and gradually adjusts its parameters; (2) The exploitation step, where new samples are presented to the system to perform a generalization [14].

These approaches suffer from difficulty in following the evolution of the user's interests. Some works found in the literature [3, 11] address this problem as a need for balancing exploration and exploitation studied in the "bandit algorithm". A bandit algorithm B exploits its past experience to select documents that appear more frequently. Besides, these seemingly optimal documents may in fact be suboptimal, due to imprecision in B's knowledge. In order to avoid this undesired situation, B has to explore documents by actually choosing seemingly suboptimal documents so as to gather more information about them. Exploitation can decrease short-term user's satisfaction since some suboptimal documents may be chosen. However, obtaining information about the documents' average rewards (i.e., exploration) can refine B's estimate of the documents' rewards and in turn increase long-term user's satisfaction. Clearly, neither a purely exploring nor a purely exploiting algorithm works best in general, and a good tradeoff is needed. The authors on [3, 11] describe a smart way to balance exploration and exploitation in the field of recommender systems. However, none of them consider the user's situation during the recommendation.

2.2 Managing the User's Situation

Few research works are dedicated to manage the user's situation on recommendation. In [1, 4,5] the authors propose a method which consists of building a dynamic situation and user profile based on time and user's experience. The user's preferences and interests in the user profile are weighted according to the situation (time, location) and user behavior. To model the change on user's preferences according to his temporal situation in different periods, like workday or vacations, the weighted association for the concepts in the user profile is established for every new experience of the user. The user activity combined with the user profile are used together to filter and recommend relevant content.

Another work [2] describes a MCRS operating on three dimensions of context that complement each other to get highly targeted. First, the MCRS analyzes information such as clients' address books to estimate the level of social affinity among users. Second, it combines social affinity with the spatiotemporal dimensions and the user's history in order to improve the quality of the recommendations.

Each work cited above tries to recommend interesting information to users on contextual situation; however they do not consider the evolution of the user's interest.

To summarize, none of the mentioned works tackles both problems. This is precisely what we intend to do with our approach, exploiting the following new features:

- Inspired by models of human reasoning developed by [7] in robotic, we propose to consider the user's situation in the bandit algorithm by using the case-based reasoning technique, which is not considered in [3, 4, 14].
- In [3, 14] authors use a smart bandit algorithm to manage the exploration/exploitation strategy, however they do not take into account the content in the strategy. Our intuition is that, considering the content when managing the exploration/exploitation strategy will improve it. This is why we propose to use content-based filtering techniques together with ε-greedy algorithm.

In what follows, we summarize the terminology and notations used in our contribution, and then we detail our methods for inferring the recommendation.

3 The Proposed MCRS Algorithm

3.1 Terminology and Notations

User Profile. The user profile is composed of the user's personal data and other dynamic information, including his preferences, his calendar and the history of his interactions with the system.

User Preferences. Preferences are deduced during user navigation activities. They contain the set of navigated documents during a situation. A navigation activity expresses the following sequence of events: (i) the user logs in the system and navigates across documents to get the desired information; (ii) the user expresses his/her preferences on the visited documents. We assume that a visited document is relevant, and thus belongs to the user's preferences, if there are some observable user's behaviors through 2 types of preference:

- The direct preference: the user expresses his interest in the document by inserting a rate, like for example putting stars ("*") at the top of the document.
- The indirect preference: it is the information that we extract from the user system interaction, for example the number of clicks or the time spent on the visited documents.

Let UP be the preferences submitted by a specific user to the system at a given situation. Each document in UP is represented as a single vector d=(c_1,...,c_n), where c_i (i=1, .., n) is the value of a component characterizing the preferences of d. We consider the following components: the total number of clicks on d, the total time spent reading d, the number of times d was recommended, and the direct preference rate on d.

History. All the interactions between the user and the system are stored together with the corresponding situations in order to exploit this data to improve the recommendation process.

Calendar. The user's calendar has information concerning the user's activities, like meetings. Time and location information is automatically inferred by the system.

User Situation. A situation *S* is represented as a triple whose features *X* are the values assigned to each dimension: $S = (X_l, X_t, X_s)$, where X_l (resp. X_t and X_s) is the value of the location (resp. time and social) dimension.

Suppose the user is associated to: the location "48.8925349, 2.2367939" from his phone's GPS; the time "Mon Oct 3 12:10:00 2011" from his phone's watch; and the meeting with Paul Gerard from his calendar. To build the situation, we associate to this kind of low level data, directly acquired from mobile devices capabilities, more abstracted concepts using ontologies reasoning means.

- **Location:** We use a local spatial ontology to represent and reason on geographic information. Using this ontology, for the above example, we get, from location "48.8925349, 2.2367939", the value “Paris” to insert in the location dimension of the situation.
- **Time:** To allow a good representation of the temporal information and its manipulation, we propose to use OWL-Time ontology [6] which is today a reference for representing and reasoning about time. We propose to base our work on this ontology and extend it if necessary. Taking the example above, for the time value "Mon Oct 3 12:10:00 2011", we get, using the OWL-Time ontology, the value “workday”.
- **Social connection:** The social connection refers to the information of the user's interlocutors (e.g. a friend, an important customer, a colleague or his manager). We use the FOAF Ontology [9] to describe the social network by a set of concepts and properties. For example, the information about “the meeting with Paul Gerard” can yield the value “wine client” for the social dimension.

3.2 The Bandit Algorithm

In our MCRS, documents' recommendation is modeled as a multi-armed bandit problem. Formally, a bandit algorithm proceeds in discrete trials t = 1,…T. For each trial t, the algorithm performs the following tasks:

- Task 1. It observes the current user u_t and a set A_t of arms together with their feature vectors $x_{t,a}$ for a ∈ A_t. The vector $x_{t,a}$ summarizes information of both user u_t and arm a, and is referred to as the context.

- Task 2. Based on observed rewards in previous trials, it chooses an arm $a_t \in A_t$, and receives reward r_{t,a_t} whose expectation depends on both the user u_t and the arm a_t.
- Task 3. It improves its arm-selection strategy with the new observation, $(x_{t,a_t}, a_t, r_{t,a_t})$. It is important to emphasize here that no feedback (namely the reward $r_{t,a}$) is observed for unchosen arms $a \neq a_t$.

In tasks 1 to 3, the total T-trial reward of A is defined as $\sum_{t=1}^{T} r_{t,a_t}$ while the optimal expected T-trial reward is defined as $\mathrm{E}\left[\sum_{t=1}^{T} r_{t,a_t^*}\right]$where a_t^* is the arm with maximum expected reward at trial t. Our goal is to design the bandit algorithm so that the expected total reward is maximized.

In the field of document recommendation, we may view documents as arms. When a document is presented to the user and this one selects it by a click, a reward of 1 is incurred; otherwise, the reward is 0. With this definition of reward, the expected reward of a document is precisely its Click Through Rate (CTR). The CTR is the average number of clicks on a recommended document, computed diving the total number of clicks on it by the number of times it was recommended. Consequently, choosing a document with maximum CTR is equivalent, in our bandit algorithm, to maximizing the total expected rewards.

3.3 The Proposed Hybrid-ε-greedy Algorithm

There are several strategies which provide an approximate solution to the bandit problem. Here, we focus on two of them: the greedy strategy, which always chooses the best arms, thus uses only exploitation; the ε-greedy strategy, which adds some greedy exploration policy, choosing the best arms at each step if the policy returns the greedy arms (probability = ε) or a random arms otherwise (probability = 1 – ε).

We propose a two-fold improvement on the performance of the ε-greedy algorithm: integrating case base reasoning (CBR) and content based filtering (CBF). This new proposed algorithm is called hybrid-ε-greedy and is described in (Alg. 3).

To improve exploitation of the ε-greedy algorithm, we propose to integrate CBR into each iteration: before choosing the document, the algorithm computes the similarity between the present situation and each one in the situation base; if there is a situation that can be re-used, the algorithm retrieves it, and then applies an exploration/exploitation strategy.

In this situation-aware computing approach, the premise part of a case is a specific situation S of a mobile user when he navigates on his mobile device, while the value part of a case is the user's preferences UP to be used for the recommendation. Each case from the case base is denoted as C= (S, UP).

Let $S^c=(X_l^c, X_t^c, X_s^c)$ be the current situation of the user, UP^c the current user's preferences and $PS=\{S^1,....,S^n\}$ the set of past situations. The proposed hybrid-ε-greedy algorithm involves the following four methods.

RetrieveCase() (Alg. 3)

Given the current situation S^c, the RetrieveCase method determines the expected user preferences by comparing S^c with the situations in past cases in order to choose the most similar one S^s. The method returns, then, the corresponding case (S^s, UP^s).

S^s is selected from PS by computing the following expression as it done in [4]:

$$S^S = \arg\max_{S^i \in PS}\left(\sum_j \alpha_j \cdot sim_j\left(X_j^c, X_j^i\right)\right) \tag{1}$$

In equation 1, sim_j is the similarity metric related to dimension j between two situation vectors and α_j the weight associated to dimension j. α_j is not considered in the scope of this paper, taking a value of 1 for all dimensions.

The similarity between two concepts of a dimension j in an ontological semantic depends on how closely they are related in the corresponding ontology (location, time or social). We use the same similarity measure as [12] defined by equation 2:

$$sim_j\left(X_j^c, X_j^i\right) = 2 * \frac{deph(LCS)}{(deph(X_j^c) + deph(X_j^i))} \tag{2}$$

Here, LCS is the Least Common Subsumer of X_j^c and X_j^i, and *depth* is the number of nodes in the path from the node to the ontology root.

RecommendDocuments() (Alg. 3*)*

In order to insure a better precision of the recommender results, the recommendation takes place only if the following condition is verified: sim(S^c, S^s) ≥ B (Alg. 3), where B is a threshold value and

$$sim(S^c, S^s) = \sum_j sim_j\left(X_j^c, X_j^s\right)$$

In the RecommendDocuments() method, sketched in Algorithm 1, we propose to improve the ε-greedy strategy by applying CBF in order to have the possibility to recommend, not the best document, but the most similar to it (Alg. 1). We believe this may improve the user's satisfaction.

The CBF algorithm (Alg. 2) computes the similarity between each document $d=(c_1,..,c_k)$ from UP (except already recommended documents D) and the best document $d^b=(c_j^b$,.., c_k^b) and returns the most similar one. The degree of similarity between d and d^b is determined by using the cosine measure, as indicated in equation 3:

$$\cos sim(d, d^b) = \frac{d \cdot d^b}{\|d\| \cdot \|d^b\|} = \frac{\sum_k c_k \cdot c_k}{\sqrt{\sum_k c_k^{b2} \cdot \sum_k c_k^{b2}}} \tag{3}$$

```
Algorithm 1. The RecommendDocuments() method
Input: ε, $UP^c$, N
Output:  D
D = ∅
For i=1 to N do
     q = Random({0, 1})
     j = Random({0, 1})
           ⎧ $\text{argmax}_{d\in (UP-D)}$ (getCTR(d))                  if j<q<ε
     $d_i$ = ⎨ CBF($UP^c$-D, $\text{argmax}_{d\in (UP-D) (}$getCTR(d))    if q≤j≤ε
           ⎩ Random($UP^c$)                                    otherwise
     D = D ∪ {$d_i$}
Endfor
Return D
```

```
Algorithm 2. The CBF() method
Input: UP, $d^b$
Output: $d^s$
$d^s = \operatorname{argmax}_{d \in (UP)}(\text{cossim}(d^b, d))$
Return $d^s$
```

UpdateCase() & InsertCase().

After recommending documents with the RecommendDocuments method (Alg. 3), the user's preferences are updated w. r. t. number of clicks and number of recommendations for each recommended document on which the user clicked at least one time. This is done by the UpdatePreferences function (Alg. 3).

Depending on the similarity between the current situation S^c and its most similar situation S^s (computed with RetrieveCase()), being 3 the number of dimensions in the context, two scenarios are possible:

- sim(S^c, S^s) $\neq$ 3: the current situation does not exist in the case base (Alg. 3); the InsertCase() method adds to the case base the new case composed of the current situation Sc and the updated UP.

- sim(S^c, S^s) = 3: the situation exists in the case base (Alg. 3); the UpdateCase() method updates the case having premise situation S^c with the updated UP.

```
Algorithm 3. hybrid-ε-greedy algorithm
Input:  B, ε, N, PS, $S^s$, $UP^s$, $S^c$, $UP^c$
Output: D
D = ∅
($S^s$, $UP^s$) = RetrieveCase($S^c$, PS)
if sim($S^c$,$S^s$) ≥ B then
      D = RecommendDocuments(ε, $UP^s$, N)
      $UP^c$ = UpdatePreferences($UP^s$, D)
      if sim($S^c$, $S^s$) ≠ 3 then
             PS = InsertCase($S^c$, $UP^c$)
      else
             PS = UpdateCase($S^p$, $UP^c$)
      end if
else     PS = InsertCase($S^c$, $UP^c$);
end if
Return D
```

4 Experimental Evaluation

In order to empirically evaluate the performance of our algorithm, and in the absence of a standard evaluation framework, we propose an evaluation framework based on a diary study entries. The main objectives of the experimental evaluation are: (1) to find the optimal threshold B value of step 2 (Section 3.3) and (2) to evaluate the performance of the proposed hybrid ε-greedy algorithm (Alg. 3) w. r. t. the optimal ε value and the dataset size. In the following, we describe our experimental datasets and then present and discuss the obtained results.

4.1 Experimental datasets

We conducted a diary study with the collaboration of the French software company Nomalys. To allow us conducting our diary study, Nomalys decides to provide the “Ns” application of their marketers a history system, which records the time, current location, social information and the navigation of users when they use the application during their meetings (social information is extracted from the users’ calendar).

The diary study took 8 months and generated 16 286 diary situation entries. Table 1 illustrates three examples of such entries where each situation is identified by IDS.

Table 1. Diary situation entries

IDS	Users	Time	Place	Client
1	Paul	11/05/2011	75060 Paris	NATIXIS
2	Fabrice	15/05/2011	59100 Roubaix	MGET
3	Jhon	19/05/2011	75015 Paris	AUNDI

Each diary situation entry represents the capture, for a certain user, of contextual information: time, location and social information. For each entry, the captured data are replaced with more abstracted information using the ontologies. For example the situation 1 becomes as shown in Table 2.

Table 2. Semantic diary situation

IDS	Users	Time	Place	Client
1	Paul	Workday	Paris	Finance client
2	Fabrice	Workday	Roubaix	Social client
3	Jhon	Holiday	Paris	Telecom client

From the diary study, we obtained a total of 342 725 entries concerning user navigation, expressed with an average of 20.04 entries per situation. Table 3 illustrates an example of such diary navigation entries. For example, the number of clicks on a document (Click), the time spent reading a document (Time) or his direct interest expressed by stars (Interest), where the maximum stars is five.

Table 3. Diary navigation entries

IdDoc	IDS	Click	Time	Interest
1	1	2	2’	**
2	1	4	3’	***
3	1	8	5’	*****

4.2 Finding the Optimal B Threshold Value

In order to evaluate the precision of our technique to identify similar situations and particularly to set out the threshold similarity value, we propose to use a manual classification as a baseline and compare it with the results obtained by our technique. So, we manually group similar situations, and we compare the manual constructed groups with the results obtained by our similarity algorithm, with different threshold values.

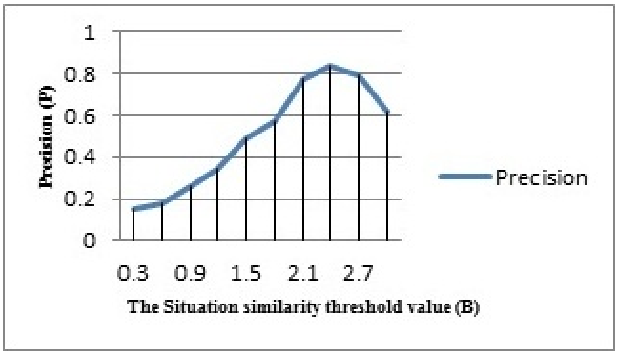


Fig. 1. Effect of B threshold value on the similarity accuracy

Figure 1 shows the effect of varying the threshold situation similarity parameter B in the interval [0, 3] on the overall precision P. Results show that the best performance is obtained when B has the value 2.4 achieving a precision of 0.849. Consequently, we use the identified optimal threshold value (B = 2.4) of the situation similarity measure for testing effectiveness of our MCRS presented below.

4.3 Experimental Datasets

In this Section, we evaluate the following algorithms: ε-greedy and hybrid-ε-greedy, described in Section 3.3; CBR-ε-greedy, a version of the hybrid-ε-greedy algorithm without executing the CBF.

We evaluated these algorithms over a set of similar user situations using the optimal threshold value identified above (B = 2.4).

The testing step consists of evaluating the algorithms for each testing situation using the traditional precision measure. As usually done for evaluating systems based on machine learning techniques, we randomly divided the entries set into two subsets. The first one, called “learning subset”, consists of a small fraction of interaction on which the bandit algorithm is run to learn/estimate the CTR associated to each document. The other one, called “deployment subset”, is the one used by the system to greedily recommend documents using CTR estimates obtained from the learning subset.

4.4 Results for ε Variation

Each of the competing algorithms requires a single parameter ε. Figures 2 and 3 show how the precision varies for each algorithm with the respective parameters. All the results are obtained by a single run.

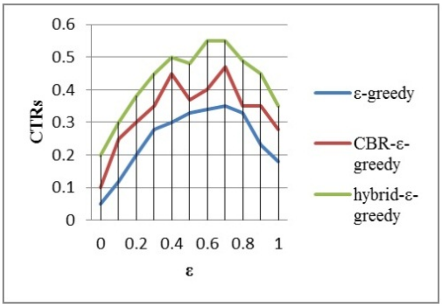


Fig. 2. ε Variation on learning subset

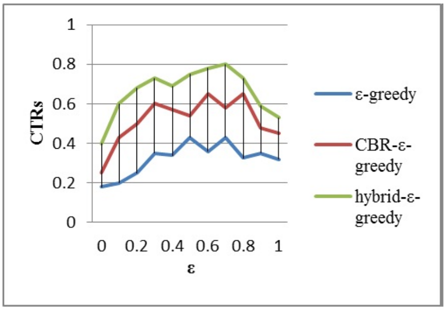


Fig. 3. ε variation on deployment subset

As seen from these figures, when the parameter ε is too small, there is insufficient exploration; consequently the algorithms failed to identify relevant documents, and had a smaller number of clicks. Moreover, when the parameter is too large, the algorithms seemed to over-explore and thus wasted some of the opportunities to increase the number of clicks. Based on these results, we choose appropriate parameters for each algorithm and run them once on the evaluation data.

We can conclude from the plots that CBR information is indeed helpful for finding a better match between user interest and document content. The CBF also helps hybrid-ε-greedy in the learning subset by selecting more attractive documents to recommend.

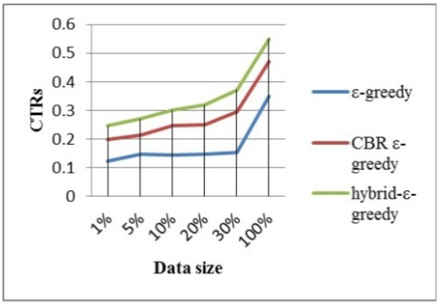


Fig. 4. Learning data size

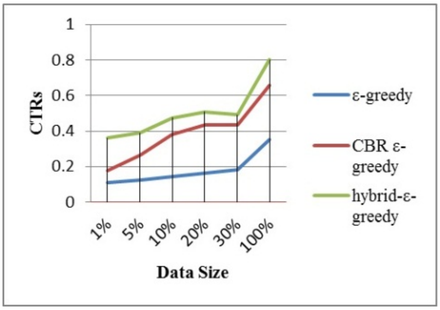


Fig. 5. Deployment data size

4.5 Valuate Sparse Data

To compare the algorithms when data is sparse in our experiments, we reduced data sizes of 30%, 20%, 10%, 5%, and 1%, respectively.

To better visualize the comparison results, figures 4 and 5 show algorithms' precision graphs with the previous referred data sparseness levels. Our first conclusion is that, at all data sparseness levels, the three algorithms are useful. A second interesting conclusion is that hybrid-ε-greedy's methods outperform the ε-greedy's one in learning and deployment subsets. The advantage of hybrid-ε-greedy over ε-greedy is even more apparent when data size is smaller. At the level of 1% for instance, we observe an improvement of 0.189 in hybrid-ε-greedy's precision using the deployment subset (0.363) over the ε-greedy's one (0.174).

5 Conclusion

This paper describes our approach for implementing a MCRS. Our contribution is to make a deal between exploration and exploitation for learning and maintaining user's interests based on his/her navigation history.

We have presented an evaluation protocol based on real mobile navigation. We evaluated our approach according to the proposed evaluation protocol. This study yields to the conclusion that considering the situation in the exploration/exploitation strategy significantly increases the performance of the recommender system following the user interests.

In the future, we plan to compute the weights of each context dimension and consider them on the detection of user's situation, and then we plan to extend our situation with more context dimension. Regarding the bandit algorithms we plan to investigate methods that automatically learn the optimal exploitation and exploration tradeoff.

References

1. Bellotti, V., Begole, B., Chi, E.H., Ducheneaut, N., Fang, J., Isaacs, E., King, T., Newman, M.W., Walendowski, A.: Scalable architecture for context-aware activity-detecting mobile recommendation system. In: 9th IEEE International Symposium on a World of Wireless, WOWMOM 2008 (2008)
2. Lakshmish, R., Deepak, P., Ramana, P., Kutila, G., Dinesh, G., Karthik, V., Shivkumar, K.: A Mobile context-aware, Social Recommender System for Low-End Mobile Devices. In: Tenth International Conference on Mobile Data Management: CAESAR 2009, pp. 338–347 (2009)
3. Lihong, L., Wei, C., Langford, J., Schapire, R.E.: A Contextual-Bandit Approach to Personalized News Article Recommendation. Presented at the Nineteenth International Conference on World Wide Web, CoRR 2010, Raleigh, vol. abs/1002.4058 (2010)
4. Bouidghaghen, O., Tamine-Lechani, L., Boughanem, M.: Dynamically Personalizing Search Results for Mobile Users. In: Andreasen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) FQAS 2009. LNCS, vol. 5822, pp. 99–110. Springer, Heidelberg (2009)
5. Panayiotou, C., Maria, I., Samaras, G.: Using time and activity in personalization for the mobile user. In: On Fifth ACM International Workshop on Data Engineering for Wireless and Mobile Access, MobiDE 2006, pp. 87–90 (2006)
6. Peter, S., Linton, F., Joy, D.: OWL: A Recommender System for Organization-Wide Learning. In: Educational Technology, ET 2000, vol. 3, pp. 313–334 (2000)
7. Bianchi, R.A.C., Ros, R., Lopez de Mantaras, R.: Improving Reinforcement Learning by Using Case Based Heuristics. In: McGinty, L., Wilson, D.C. (eds.) ICCBR 2009. LNCS, vol. 5650, pp. 75–89. Springer, Heidelberg (2009)

8. Samaras, G., Panayiotou, C.: Personalized Portals for the Wireless and Mobile User. In: Proc. 2nd Int'l Workshop on Mobile Commerce, ICDE 2003, p. 792 (2003)
9. Shijun, L., Zhang, Y., Xie, Sun, H.: Belief Reasoning Recommendation Mashing up Web Information Fusion and FOAF:JC 2010. Journal of Computers, JC 02(12), 1885–1892 (2010)
10. Varma, V., Kushal, S.: Pattern based keyword extraction for contextual advertising. In: Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM (2010)
11. Wei, L., Wang, X., Zhang, R., Cui, Y., Mao, J., Jin, R.: Exploitation and Exploration in a Performance based Contextual Advertising System. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2010. ACM (2010)
12. Wu, Z., Palmer, M.: Verb Semantics and Lexical Selection. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, ACL 1994, pp. 133-138 (1994)
13. Zhang, T., Iyengar, V.: Recommender systems using linear classifiers. The Journal of Machine Learning Research, JMLR 2, 313–334 (2002)

Unsupervised Multi-label Text Classification Using a World Knowledge Ontology

Xiaohui Tao[1], Yuefeng Li[2], Raymond Y.K. Lau[3], and Hua Wang[1]

[1] Centre for Systems Biology, University of Southern Queensland, Australia
{xtao,hua.wang}@usq.edu.au
[2] Science and Engineering Faculty, Queensland University of Technology, Australia
y2.li@qut.edu.au
[3] Department of Information Systems, City University of Hong Kong, Hong Kong
raylau@cityu.edu.hk

Abstract. The development of text classification techniques has been largely promoted in the past decade due to the increasing availability and widespread use of digital documents. Usually, the performance of text classification relies on the quality of categories and the accuracy of classifiers learned from samples. When training samples are unavailable or categories are unqualified, text classification performance would be degraded. In this paper, we propose an unsupervised multi-label text classification method to classify documents using a large set of categories stored in a world ontology. The approach has been promisingly evaluated by compared with typical text classification methods, using a real-world document collection and based on the ground truth encoded by human experts.

1 Introduction

The increasing availability of documents in the past decades has greatly promoted the development of information retrieval and organising systems, such as search engines and digital libraries. The widespread use of digital documents has also increased these systems' accessibility to textual information. A fundamental theory supporting these information retrieval and organising systems is that information can be associated with semantically meaningful categories. Such a theory supports also ontology learning, text categorisation, information filtering, text mining, and text analysis, etc. Text classification aims at associating textual documents with semantically meaningful categorises, and has been studied in the past decades, along with the development of information retrieval and organising systems [11].

Text classification is the process of classifying an incoming stream of documents into predefined categories. Text classification usually employs a supervised learning strategy with the classifiers learned from pre-classified sample documents. The classifiers are then used to classify incoming documents. In terms of supervised text classification, the performance is determined by the accuracy of pre-classified training samples and the quality of the categorisation. The accuracy of classifiers determines their capability of differentiating the incoming

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 480–492, 2012.

stream of documents; the descriptive and discriminative capacity of categorisation reduces noise in classification, which is caused by sense ambiguities, sparsity, and high dimensionality of the documents [7]. Text classification performance is also affected by the topic coverage of categories. An inadequate category may be assigned to a document if an in-comprehensive set of categories is employed, because non-adequate categories can be found. The performance of text classification relies upon the descriptive and discriminative capacity of categories and the accuracy of classifiers learned from training sets.

However, there exist situations that a qualified training document set may not be available (e.g., the "cold start" problem in recommender systems); a set of categories with in-comprehensive topic coverage may be used for classification; sometimes although a set of categories with comprehensive topic coverage is available, the large number of classes would easily introduce noise in classification results [5]. Traditionally, text classification models are designed to handle only single-label problems. However, in some circumstances (e.g., categorizing documents in library catalogue into multiple subjects), multi-label text classification is required and automatic classification is necessary, especially when classifying a very large volume of documents [15]. To deal with these problems, in this paper we propose an automatic unsupervised text classification approach to classify documents into multiple classes, without the requirement of pre-classified sample documents for training classifiers. The approach consists of three modules; pattern mining for document feature extraction; feature-subject mapping for initial classification; knowledge generalisation for optimal classification. The method incorporates comprehensive world knowledge stored in a large ontology and classifies documents into the classes in the ontology without any pre-classified training samples available. The world ontology is built from Library of Congress Subject Headings (LCSH), which represents the natural growth and distribution of human intellectual work [4]. The subject classes and semantic relationships in the ontology are investigated and exploited to improve the classification results. The proposed method was experimentally evaluated using a large library catalogue, by compared with typical text classification approaches. The presented work makes three-fold contributions:

- An unsupervised text classification method that classifies documents into multiple classes;
- A knowledge generalisation method to optimise text classification by analysing the semantic relations of categories;
- An exploration of using the LCSH as a world knowledge to facilitate text classification.

The paper is organised as follows. Section 2 discusses the related work; Section 3 introduces the research problem and the the conceptual model of proposed supervised text classification method; Section 4 presents the technical detail of the proposed method. The experiment design is described in Section 5, whereas the results are discussed in Section 6. Finally, Section 7 makes conclusions.

2 Related Work

Unsupervised text classification aims to classify documents into the classes with absence of any labelled training documents. In many occasions the target classes may not have any labelled training documents available. One particular example is the "cold start" problem in recommender systems and social tagging. Unsupervised classification can automatically learn an annotation model to make recommendations or label the tags when the products or tags are rare and do not have any useful information associated. Unsupervised classification has been studied by many groups and many successful models have been proposed. Without associated training samples, Yang et al. [16] built a classification model for a target class by analysing the correlating auxiliary classes. Though as similar as theirs in investigating correlating classes, our work is different by exploiting a hierarchical world knowledge ontology for classification, instead of only auxiliary classes. Also exploiting a world knowledge base, Yan et al. [14] examined unsupervised relation extraction from Wikipedia articles and integrated linguistic analysis with web frequency information to improve unsupervised classification performance. However, our work has different aims from theirs; ours aims to exploit a world knowledge ontology to help unsupervised classification, whereas Yan et al. [14] aims to extract semantic relations for Wikipedia concepts by using unsupervised classification techniques. Cai et al. [2] and Houle and Grira [6] proposed unsupervised approaches to evaluate and improve the quality of selecting features. Given a set of data, their work is to find a subset containing the most informative, discriminative features. Though the work presented in this paper also relies on features selected from documents, the features are further investigated with their referring-to ontological concepts to improve the performance of classification.

Text classification models are originally designed to handle only single-label problems, where each document is classified into only one class. However, in many circumstances single-label text classification cannot satisfy the demand, for example, in social network multiple labels may need to be suggested for a tag [8]. Comparing with the work done by Katakis et al. [8], our work relies on the semantic content of documents, rather than the meta-data of documents used in [8]. As similar as the work conducted by Yang et al. [15], our work also targets on multi-label text classification. However, Yang et al. [15]' work is different in adopting active learning algorithms for multi-label classification, whereas ours exploits concepts and their structure in world knowledge ontologies.

Ontologies have been studied and exploited by many works to facilitate text classification. Gabrilovich and Markovitch [5] enhanced text classification by generating features using domain-specific and common-sense knowledge in large ontologies with hundreds of thousands of concepts. Comparing with their work, our work moves beyond feature discovery and investigates the hierarchical ontology structure for knowledge generalisation to improve text classification. Camous et al. [3] also introduced a domain-independent method that uses the Medical Subject Headings (MeSH) ontology. The method observes the inter-concept relationships and represents documents by MeSH subjects. Similarly, Camous' work considers the semantic relations existing in the ontological concepts. However, their work focuses on only the medical domain, whereas our approach works on general areas because exploiting the LCSH, a superior world knowledge ontology. Another world ontology commonly used in text classification is Wikipedia. Wang and Domeniconi [13] and Hu et al. [7] derived background knowledge from Wikipedia to represent documents and attempted to deal with the sparsity and high dimensionality problems in text classification. Instead of Wikipedia with free-contributed entries, our work uses the superior LCSH ontology, which has been under continuous development for a hundred years by knowledge engineers.

Many works utilise pattern mining techniques to help build classification models, which is similar as the strategy employed in our work. Malik and Kender [10] proposed the "Democratic Classifier", which is a pattern-based classification algorithm using short patterns. Different from our work, their democratic classifier relies on the quality of training samples and cannot deal with the "no training set available" problem. Bekkerman and Matan [1] argued that most of information on documents can be captured in phrases and proposed a text classification method that employs lazy learning from labelled phrases. The phrases in their work are in fact a special form of sequential patterns that are used in our work for feature extraction of documents.

3 Unsupervised Multi-label Text Classification

Let $\mathcal{D} = \{d_i \in \mathbb{D}, i = 1, \ldots, m\}$ be a set of text documents; $\mathcal{S} = \{s_1, \ldots, s_K\}$ be a large set of classes, where K is the number of classes. If there is available a training set $\mathcal{D}_t = \{d_j \in \mathbb{D}, j = m+1, \ldots, n\}$ with $y_j^k = \{0, 1\}, k = 1, \ldots, K$ provided for describing the likelihood of d_j belonging to class s_k, it is easy to learn a binary prediction function $p(y^k|d)$ and use it to classify $d_i \in \mathcal{D}$. However, our objective is to learn a prediction function $p(y^k|d)$ to classify d_i into $\{s_k\} \subset \mathcal{S}$ without $\mathcal{D}_t$ available. We refer to this problem as *unsupervised multi-label text classification*.

The proposed classification method consists of three steps: feature extraction, initial classification, and optimising classification, using a world ontology.

3.1 World Ontology

The world knowledge ontology is constructed from the Library of Congress Subject Headings (LCSH), which is a knowledge system developed for organising information in large library collections. It has been under continuous development for over a hundred years to describe and classify human knowledge. Because of the endeavours dedicated by the knowledge engineers from generation to generation, the LCSH has become a de facto standard for concept cataloguing and indexing, superior to other knowledge bases. Tao et al. [12] once compared the LCSH with the Library of Congress Classification, the Dewey Decimal Classification, and Yahoo! categorisation, and reported that the LCSH has broader topic coverage, more meaningful structure, and more accurate semantic relations. The LCSH has been widely used as a means for many knowledge engineering and management works [4]. In this work, the class set $\mathcal{S} = \{s_1, \ldots, s_K\}$ is encoded from the LCSH subject headings.

Definition 1. (SUBJECT) *Let $\mathcal{S}$ be the set of subjects, an element $s \in \mathcal{S}$ is a 4-tuple $s := \langle label, neighbour, ancestor, descendant \rangle$, where*

- *label is a set of sequential terms describing s; $lable(s) = \{t_1, t_2, \ldots, t_n\}$;*
- *neighbour refers to the set of subjects in the LCSH that directly link to s, $neighbour(s) \subset \mathcal{S}$;*
- *ancestor refers to the set of subjects directly and indirectly link to s and locating at more abstractive level than s in the LCSH, $ancestor(s) \subset \mathcal{S}$;*
- *descendant refers to the set of subjects directly and indirectly link to s and locating at more specific level than s in the LCSH, $descendant(s) \subset \mathcal{S}$.* □

The semantic relationships of subjects are encoded from the references defined in the LCSH for subject headings, including *Broader Term*, *Used for*, and *Related to*. The $ancestor(s)$ in Definition 1 returns the *Broader Term* subjects of s; the $descendant(s)$ is the reversed function of $ancestor(s)$, with additional subjects *Used for s*; the $neighbour(s)$ returns the subjects *Related to s*.

With Definition 1, the world knowledge ontology is defined:

Definition 2. (ONTOLOGY) *Let $\mathcal{O}$ be a world ontology. $\mathcal{O}$ contains a set of subjects linked by their semantic relations in a hierarchical structure. $\mathcal{O}$ is a 3-tuple $\mathcal{O} := \langle \mathcal{S}, \mathcal{R}, \mathcal{H}_{\mathcal{R}}^{\mathcal{S}} \rangle$, where*

- *$\mathcal{S}$ is the set of subjects defined in Definition 1;*
- *$\mathcal{R}$ is the set of relations linking any pair of subjects;*
- *$\mathcal{H}_{\mathcal{R}}^{\mathcal{S}}$ is the hierarchical structure of $\mathcal{O}$ constructed by $\mathcal{S} \times \mathcal{R}$.* □

3.2 Document Features

Various representations have been studied to formally describe text documents. The lexicon-based representation is based on the statistic of occurring terms. Such a representation is easy to understand by users and systems. However, along with meaningful, representative features, some noisy terms are also extracted, caused by sense ambiguity of terms. To deal with this problem, pattern-based representation is studied, which uses frequent sequential patterns (phrases) to represent document contents [9]. The pattern-based representation is superior to lexicon-based, as the context of terms co-occurred in phrases is considered. However, the pattern-based presentation suffers from a limitation caused by the length of patterns. Though a long pattern is wealthy with information and so more discriminative, it usually has low frequency and as a result, becomes inapplicable. To overcome the problem, we represent the content of documents by a set of weighted closed frequent sequential patterns discovered by pattern mining techniques.

Definition 3. (FEATURES) *Given a document $d = \{t_1, t_2, \ldots, t_n\}$ as a sequential set of repeatable terms, the feature set, denoted as $\mathcal{F}(d)$, is a set of weighted phrase patterns, $\{\langle p, w(p) \rangle\}$, extracted from d that satisfies the following constraints:*

- $\forall p \in \mathcal{F}(d), p \subseteq d$.
- $\forall p_1, p_2 \in \mathcal{F}(d)(p_1 \neq p_2), p_1 \not\subset p_2 \wedge p_2 \not\subset p_1$.
- $\forall p \in \mathcal{F}(d), w(p) \geqslant \vartheta$, *a threshold.* □

3.3 Initial Classification

The initial classification of d to $s_k \in \mathcal{S}$ is done through accessing a term-subject matrix created by the subjects and their labels. Adopting the features discovered previously, we use a feature-subject mapping approach to initially assign subject classes to the document.

Definition 4. (TERM-SUBJECT MATRIX) *Let $\mathcal{T}$ be the term space of $\mathcal{S}, \mathcal{T} = \{t \in \bigcup_{s \in \mathcal{S}} label(s)\}, \langle \mathcal{S}, \mathcal{T} \rangle$ is the matrix coordinated by $\mathcal{T}$ and $\mathcal{S}$, where a mapping exists:*

$$\mu : \mathcal{T} \to 2^{\mathcal{S}}, \quad \mu(t) = \{s \in \mathcal{S} | t \in label(s)\}$$

and its reverse mapping also exists:

$$\mu^{-1} : \mathcal{S} \to 2^{\mathcal{T}}, \;\; \mu^{-1}(s) = \{t \in \mathcal{T} | s \in \mu(t)\}$$ □

Adopting Definition 3 and 4, we can initially classify $d_i \in \mathcal{D}$ into a set of subjects using the following prediction:

$$\widehat{y}_i^k = I(s_k \in h \circ g \circ f(d_i)), i = 1, \ldots, m \tag{1}$$

where $I(z)$ is an indicator function that outputs 1 if z is true and zero, otherwise; $f(d) = \{p | \langle p, w(p) \rangle \in F(d)\}$; $g(\rho) = \{t \in \cup_{p \in \rho} p\}$; $h(\tau) = \{s \in \cup_{t \in \tau} \mu(t)\}$.

3.4 Generalised Classification

The initial classification process easily generates noisy subjects because of direct feature-subject mapping. Against the problem, we introduce a method to generalise the initial subjects to optimise the classification. We observed that in initial classification some subjects extracted from the ontology are overlapping in their semantic space. Thus, we can optimise the classification result by keeping only the dominating subjects and pruning away those being dominated. This can be done by investigating the semantic relations existing between subjects. Let s_1 and s_2 be two subjects and $s_1 \in ancestor(s_2)$ ($s_2 \in descendant(s_1)$). s_1 refers to an broader semantic space than s_2 and thus, is more general. Vice versa, s_2 is more specific and focused than s_1. Hence, if some subjects are covered by a common ancestor, they can be replaced by the common ancestor without information loss. The common ancestor is unnecessary to be chosen from the initial classification result, as choosing an external common ancestor also satisfies the above rule. After generalising the initial classification result, we have a smaller set of subject classes, with no information lost but some focus. (The handling of focus problem is presented in next section.)

```
input : $d = \{t_1, t_2, \ldots, t_n\}$ where $n = |d|$, a threshold $\vartheta$.
output: The feature set $\mathcal{F}(d) = \{\langle p, w(p)\rangle\}$.
$P(d) = \emptyset, \mathcal{F}(d) = \emptyset, p = \emptyset$;
//Extracting sequential patterns;
for $(i = 1; i <= n; i++)$ do
    for $(j = i; j <= (n - i); j++)$ do
        $p = p \cup \{t_j\}$;
    end
    if $p \in P(d)$ then $w(p)++$ for $\langle p, w(p)\rangle \in \mathcal{F}(d)$else $P(d) = P(d) \cup \{p\}$,
    $\mathcal{F}(d) = \mathcal{F}(d) \cup \{\langle p, 1\rangle\}$;
end
//Filtering $\mathcal{F}(d)$ for closed, frequent patterns;
foreach $\langle p, w(p)\rangle \in \mathcal{F}(d)$ do
    if $w(p) < \vartheta$ then $\mathcal{F}(d) = \mathcal{F}(d) - \{\langle p, w(p)\rangle\}$else foreach $\langle p_k, w(p_k)\rangle \in \mathcal{F}(d)$ do
        if $p \subset p_k$ and $w(p) \leq w(p_k)$ then $\mathcal{F}(d) = \mathcal{F}(d) - \{\langle p, w(p)\rangle\}$
    end
end
return $\mathcal{F}(d)$.
```

Algorithm 1. Extracting Features from a Document

Definition 5. (GENERALISED CLASSIFICATION) *Given a document d and its initial classification result, a subject set denoted by $S^I(d)$, the generalised classification result, denoted as $S^G(d)$, is the set of subjects satisfying:*

1. $\forall s \in S^I(d), \exists s' \in S^G(d), s \neq s', s \in descendants(s')$.
2. $\forall s_1, s_2 \in S^G(d)(s_1 \neq s_2), s_1 \notin descendants(s_2) \wedge s_2 \notin descendants(s_1)$.

4 Implementation

In this section, we present the technical details for implementing the proposed approach of unsupervised multi-label text classification.

Algorithm 1 describes the process of extracting features to represent a document. The output is $\mathcal{F}(d)$, a set of closed frequent sequential patterns discovered from d. Adopting the prediction in Eq. (1), with $\mathcal{F}(d)$ the initial set of subjects, $S^I(d)$, can be assigned to classify d. Taking into account the weights of feature patterns, we can evaluate $t \in d$:

$$w(t) = \sum_{p \in \{p | t \in g \circ f(d), p \in f(d)\}} w(p)$$

All $s \in S^I(d)$ can then be re-evaluated for their likelihood of being assigning to d with consideration of term evaluation and term distribution in $s \in S^I(d)$. A prediction function can then be used to assess initial classification subjects for the second run of classification:

$$\widehat{y'}_i^{\kappa} = \mathcal{I}(\sum_{t \in \mu^{-1}(s_\kappa)} w(t) \times log(\frac{|S^I(d_i)|}{sf(t, S^I(d_i))}) \geqslant \theta), i = 1, \ldots, m \tag{2}$$

where $\mathcal{I}(z \geqslant \theta)$ returns the value of z if $z \geqslant \theta$ is true and zero, otherwise; $\kappa = 1, ..., \mathcal{K}$ and $S^I(d) = \{s_1, \ldots, s_{\mathcal{K}}\}$ with $|S^I(d)| = \mathcal{K}$; θ is the threshold for

```
  input  : $S_i = \{s_1, s_2, \ldots, s_j\}$ (subject classes assigned to $d_i$ after Eq. (2)), $\mathcal{O}$;
  output: $\mathcal{S'}_i = \{s_1, s_2, \ldots, s_k\}$ (subject classes generalised for optimising classification).
1 $\mathcal{S'}_i = \emptyset, S_{temp} = \emptyset, S_{redundant} = \emptyset$;
2 foreach $s \in S_i$ do
3     Extract $S(s)$ from $\mathcal{O}$ where $S(s) = \{s' | s' \in ancestor(s), \delta(s \mapsto s') \leq 3\}$; foreach
      $s_n \in S_i$ where $s_n \neq s$ do
4         Extract $S(s_n)$ from $\mathcal{O}$ like Step 3;
5         if $S(s) \cap S(s_n) \neq \emptyset$ then
          $\{\widehat{s} = \mathcal{LCA}(S(s) \cup S(s_n)), str(i, \widehat{s}) = str(i, s) + str(i, s_n)$; $S_{temp} = S_{temp} \cup \{\widehat{s}\}$;
          $S_{redundant} = S_{redundant} \cup \{s, s_n\}\}$
6     end
7     if $S_{temp} \neq \emptyset$ then $\{\mathcal{S'}_i = \mathcal{S'}_i \cup S_{temp}$; $S_i = S_i - S_{redundant}$; $S_{temp} = \emptyset$;
      $S_{redundant} = \emptyset\}$ else $\mathcal{S'}_i = \mathcal{S'}_i \cup \{s\}$
8 end
9 return $\mathcal{S'}_i$.
```

Algorithm 2. Generalising Subjects for Optimal Classification

filtering out noisy subjects. In experiments different values were tested for θ. The results revealed that setting θ as the top fifth z in $S^I(d_i)$, a variable rather than a static value, gave the best performance. (Refer to Section 6 for detail.)

In the generalisation phase, descendant subjects are replaced by their common ancestor subject. However, the common ancestor should not be too far away from the replaced descendants in the ontology structure. The focus will be significantly lost, otherwise. In implementation, we use only the lowest common ancestor (shortened by LCA) to replace its descendant subjects. The LCA is the common ancestor of a set of subjects, with the shortest distance to these subjects in the ontology structure. The LCA replaces descendant subjects with full information kept and minimised focus lost.

Algorithm 2 describes the process of generalising the initial subject classes to optimise classification. The function $str(i, s)$ describes the likelihood of assign s tod_i and returns the value of $\mathcal{I}(z \geqslant \theta)$ in Prediction function $\widehat{y'}_i^{\kappa}$ in Eq. (2). The function $\delta(s_1 \mapsto s_2)$ returns a positive real number indicating the distance between two subjects. Such a distance is measured by counting the number of edges travelled through from s_1 to s_2 in $\mathcal{H}_{\mathcal{R}}^{\mathcal{S}}$. The function $\mathcal{LCA}(S(s_1) \cup S(s_2))$ returns $\widehat{s}$, the LCA of s_1 and s_2. Note that $\delta(s_1 \mapsto s_2) \leq 3$, which restricts LCAs to three edges in distance. Subjects further than that in distance are too general; whereas using a highly-general subject for generalisation would severely jeopardise the focus of original subjects. (In the experiments, $\delta(s_1 \mapsto s_2) \leq 3$ and ≤ 5 were tested under the same environment in order to find a valid distance for tracking the competent LCA. The testing results revealed that as of three the distance was better.)

5 Evaluation

The experiments were performed, using a large corpus collected from the catalogue of a library using the LCSH for information organising. The title and content of each catalogue item were used to form the content of a document. The subject headings associated with the catalogue items were manually assigned by specialist librarians who were trained to specify subjects for documents without bias [4]. The documents and subjects provided an ideal ground truth in the experiments to evaluate the effectiveness of the proposed classification method. This objective evaluation methodology assured the solidity and reliability of the experimental evaluation.

The testing set was crawled from the online catalogue of library of the University of Melbourne[1]. General text pre-processing techniques, such as stopword removal and word stemming (Porter stemming algorithm), were applied to the preparation of testing set for experiments. In the experiments, we used only documents containing at least 30 terms, resulted in 31,902 documents in the testing set. Documents shorter than that could hardly provided substantial frequent patterns for feature extraction, as revealed in the preliminary experiments.

Given that the LCSH ontology contains 394,070 subjects in our implementation, the problem actually became a K-class text classification problem where $K = |\mathcal{S}| = 394,070$, a very large number. Hence, we chose two typical multi-class classification approaches, *Rocchio* and kNN, as the baseline models in the experiments.

The performance of experimental models was measured by precision and recall, the modern evaluation methods in information retrieval and organising. In terms of text classification, precision was to measure the ability of a method to assign a document with only focusing subjects, and recall the ability to assign a document with all dealing subjects.

Taking into account $K = |\mathcal{S}| = 394070$, in respect with the testing document set and the ground truth featured by the LCSH, the classification performance was evaluated by:

$$precision = \frac{|\mathcal{FT}(S_{tgt}) \cap \mathcal{FT}(S_{grt})|}{|\mathcal{FT}(S_{tgt})|} \text{ and } recall = \frac{|\mathcal{FT}(S_{tgt}) \cap \mathcal{FT}(S_{grt})|}{|\mathcal{FT}(S_{grt})|}$$

where $\mathcal{FT}(S) = \bigcup_{s \in S} \mu^{-1}(s)$ (see Definition 4); tgt referred to the target model; grt referred to the ground truth subjects.

F_1 Measure as another common method used in information organising systems was also employed in evaluation. We used *micro*-F_1, which evaluated each document's classification result first and then averaged the results for the final F_1 value. Greater F_1 values indicate better performance.

6 Results and Discussions

Naming our proposed unsupervised classification approach as the UTC model, the experiments were to compare the effectiveness performance of the UTC model to the baselines, Rocchio and kNN models. Their effectiveness performances are depicted in Fig. 1 for the number of documents with valid effectiveness (> 0), where the value axis indicates the effectiveness rate between 0 and 1; the category axis indicates the number of documents whose classification meets

[1] http://www.library.unimelb.edu.au/

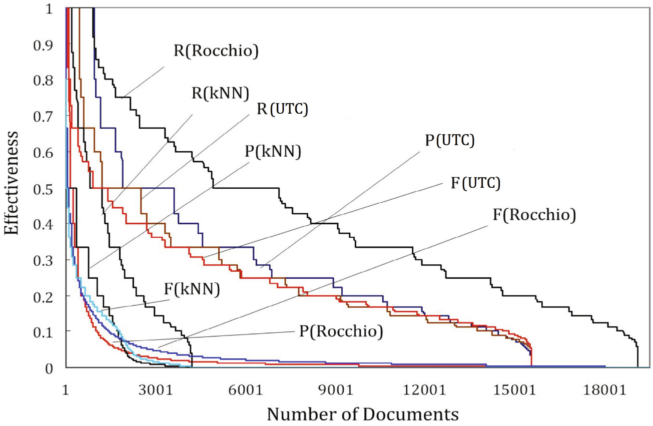


Fig. 1. Effectiveness Performance Results

Table 1. Effectiveness Performance on Average

	Precision	Recall	F-Measure
UTC	0.158	0.135	0.125
Rocchio	0.020	0.290	0.020
kNN	0.021	0.054	0.016

the respective accuracy rate. As shown in the figure, the effectiveness rates were measured by precision, recall, and F_1 Measure, where $P(x)$ refers to the precision results of experimental model x, $R(x)$ the recall results, and $F(x)$ the F_1 Measure results. Their overall average performances are shown in Table 1.

F_1 Measure equally considers both precision and recall. Thus the F_1 Measure results can be deemed as an overall effectiveness performance. The average F_1 Measure result shown in Table 1 reveals that the UTC model has achieved a much better overall performance (0.125) than other two models (0.020 and 0.016). Such a performance is also confirmed by the detailed results depicted in Fig. 1 - the $F(UTC)$ line is located at much higher bound level compared to the $F(Rocchio)$ and $F(kNN)$ lines.

Precision measures how accurate the classification is. In terms of this, the UTC model once again has outperformed the baseline models. The average precision results shown in Table 1 demonstrates the achievement (UTC 0.158 vs. Rocchio 0.020 and *k*NN 0.021). The precision results depicted in Fig. 1 illustrate the same conclusion; the $P(UTC)$ outperformed others.

Recall measures the performance of classification by considering all dealing classes. The recall performance in the experiments shows a slightly different result, compared with those from F_1 Measure and precision performance. The *Rocchio* model achieved the best recall performance (0.290 on average), compared to that of the UTC model (0.135) and the *k*NN model (0.054). The result is also illustrated in Fig. 1, where $R(UTC)$ lies in the middle of $R(Rocchio)$ and $R(kNN)$.

There was a gap between the recall performance of the UTC and the *Rocchio* models. From the observation of recall results, we found that the classes assigned by the *Rocchio* model were usually a large set of subjects (935 on average), whereas the UTC model assigned documents with a reasonable number of subjects (16 on average) and the kNN results had an average size of 106. Due to the natural of recall measurement, more feature term would be cover if the subject size became larger. As a result, the *Rocchio* classification with the largest size achieved the best recall performance. The subject sets assigned by the kNN model had larger size than those assigned by the UTC. However, when expanding the classification by neighbours, a large deal of nosey data was also brought into the neighbourhood - the average number of neighbours arisen was 336. This was caused by the very large set and short length of documents in consideration. As a result, the classification became inaccurate though only the documents with the top cosine values were chosen to expand and only the subjects with the top similarity values were chosen to classify a document.

Table 2. Performance Comparison for Finding the LCA

	Precision	Recall	F-Measure
Level = 3	0.158	0.135	0.125
Level = 5	0.154	0.112	0.111

Different number of levels were tested in sensitivity study for choosing a right number of levels to find the lowest common ancestor when generalising subjects for optimal classification. Table 2 displays the testing results for finding such a right level number. In the same experimental environment, if tracing three levels to find a LCA the UTC model's overall performance including F_1 Measure, precision, and recall was better than that of tracing five levels. In addition, tracing three levels only would give us better complexity. Therefore, we chose three levels to restrict the extent of finding CLAs.

7 Conclusions

Text classification has been widely exploited to improve the performance in information retrieval, information organising, text categorisation, and knowledge engineering. Traditionally, text classification relies on the quality of target categorises and the accuracy of classifiers learned from training samples. Sometimes qualified training samples may be unavailable; the set of categories used for classification may be with inadequate topic coverage. Sometimes documents may be classified into noisy classes because of large dimension of categories. Aiming to deal with these problems, in this paper we have introduced an unsupervised multi-label text classification method. Using a world ontology built from the LCSH, the method consists of three modules; closed frequent sequential pattern mining for feature extraction; extracting subjects from the ontology for initial classification; and generalising subjects for optimal classification. The method has been promisingly evaluated by compared with typical text classification methods, using a large real-world corpus, based on the ground truth encoded by human experts.

References

1. Bekkerman, R., Gavish, M.: High-precision phrase-based document classification on a modern scale. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2011, pp. 231–239 (2011)
2. Cai, D., Zhang, C., He, X.: Unsupervised feature selection for multi-cluster data. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2010, pp. 333–342 (2010)
3. Camous, F., Blott, S., Smeaton, A.: Ontology-Based MEDLINE Document Classification. In: Hochreiter, S., Wagner, R. (eds.) BIRD 2007. LNCS (LNBI), vol. 4414, pp. 439–452. Springer, Heidelberg (2007)
4. Chan, L.M.: Library of Congress Subject Headings: Principle and Application. Libraries Unlimited (2005)
5. Gabrilovich, E., Markovitch, S.: Feature generation for text categorization using world knowledge. In: Proceedings of The 19th International Joint Conference for Artificial Intelligence, pp. 1048–1053 (2005)
6. Houle, M.E., Grira, N.: A correlation-based model for unsupervised feature selection. In: Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management, CIKM 2007, pp. 897–900 (2007)
7. Hu, X., Zhang, X., Lu, C., Park, E.K., Zhou, X.: Exploiting wikipedia as external knowledge for document clustering. In: KDD 2009: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 389–396 (2009)
8. Katakis, I., Tsoumakas, G., Vlahavas, I.: Multilabel text classification for automated tag suggestion. In: Proceedings of the ECML/PKDD 2008 Workshop on Discovery Challenge (2008)
9. Li, Y., Algarni, A., Zhong, N.: Mining positive and negative patterns for relevance feature discovery. In: Proceedings of 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 753–762 (2010)
10. Malik, H.H., Kender, J.R.: Classifying high-dimensional text and web data using very short patterns. In: Proceedings of the 2008 8th IEEE International Conference on Data Mining, ICDM 2008, pp. 923–928 (2008)
11. Rocha, L., Mourão, F., Pereira, A., Gonçalves, M.A., Meira Jr., W.: Exploiting temporal contexts in text classification. In: Proceeding of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, pp. 243–252 (2008)
12. Tao, X., Li, Y., Zhong, N.: A personalized ontology model for web information gathering. IEEE Transactions on Knowledge and Data Engineering, IEEE Computer Society Digital Library 23(4), 496–511 (2011)
13. Wang, P., Domeniconi, C.: Building semantic kernels for text classification using wikipedia. In: KDD 2008: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 713–721 (2008)

14. Yan, Y., Okazaki, N., Matsuo, Y., Yang, Z., Ishizuka, M.: Unsupervised relation extraction by mining wikipedia texts using information from the web. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, ACL 2009, vol. 2, pp. 1021–1029 (2009)
15. Yang, B., Sun, J.-T., Wang, T., Chen, Z.: Effective multi-label active learning for text classification. In: KDD 2009: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 917–926 (2009)
16. Yang, T., Jin, R., Jain, A.K., Zhou, Y., Tong, W.: Unsupervised transfer classification: application to text categorization. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2010, pp. 1159–1168 (2010)

Semantic Social Network Analysis with Text Corpora

Dong-mei Yang[1], Hui Zheng[1], Ji-kun Yan[2], and Ye Jin[2]

[1] Science and Technology on Blind Signal Processing Laboratory
ydm.123@foxmail.com, Zheng5739@163.com
[2] Southwest Electronics and Telecommunication Technology Research Institute
yanjk@126.com, djws_1982@sina.com

Abstract. We present the Document-Entity-Topic (DET) model for semantic social network analysis which tries to find out the interested entities through the topics we aim at, detect groups according to the entities which concern the similar topics, and rank the plentiful entities in a document to figure out the most valuable ones. DET model learns the topic distributions by the literal descriptions of entities. The model is similar to Author-Topic (AT) model, adding the key attribute that the distribution of entities in a document is not uniform but Dirichlet allocation. We experiment on the "Libya Event" data set which is collected from the Internet. DET model increases the precision on tasks of social network analysis and gives much lower perplexity than AT model.

Keywords: Semantic Social Network Analysis, Topic Model, Entity Modeling.

1 Introduction

As far as we know, topic modeling has become a most popular technology to model large collection of corpus[1-3], such as Latent Dirichlet Allocation[4]. The basic idea of topic modeling is that the latent topics can be used to describe the relationship between words and documents. In this paper we consider the problem of using latent topics to connect the words and entities in documents (such as person, location, organization). We focus on the news articles which contain lots of entities in order to convey the information about who, what, when and where. The purpose we want most is modeling the entities in terms of latent topics so that we can 1) find out the interested entities through the topics we aim at; 2)recognize groups with supposing that the entities (especially the persons) which concern the similar topics can be seen as a group; 3) rank the plentiful entities in a document to figure out the valuable ones by assuming that the more an entity contributes to a document's topic(s), the more valuable it is in the precise one. We call the three tasks Semantic Social Network Analysis for the interactions been found based on the topics of the corpus.

There are several related researches to achieve the relationship between words and entities (authors) with topic models. The Author-Topic (AT) model[5-6] learns the topics of a document conditioned on the mixture of interests with the authors. AT model assumes that the authors equally contribute to the topics of a document. The SwitchLDA and GESwitchLDA[7-8] extend LDA to capture dependencies between entities and topics, referring to entities as additional classes of words.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 493–504, 2012.

This paper presents the Document-Entity-Topic (DET) model, a directed graphical model by assuming that words were generated by the entities of the document. The model is similar to the AT model. However, it is not limited to the topic finding of authors, but tries to modeling topics of all related entities in the documents. For this application, we confront more unwanted entities of the corpus. In our experiments, there are more than five person entities in most documents, and some entities such as news reporters have little significance to the topic(s). If all entities in a document have been assumed to be equally contributed to the mixture of topics as AT model, it is not enough for us to rank the importance of entities and many noisy entities will disturb the topic modeling of corpus. So our DET model presumes that the entities have different topical contributions to their document. We use the Dirichlet allocation to describe the distribution between document and its entities; a document gives higher probabilities to several more valuable entities (not all entities) and valuable entities have more contributes to the topic modeling.

The outline of the paper is as follows: Section 2 describes the Document-Entity-Topic model, and section 3 outlines how to learn the parameters from the documents. Section 4 discusses the application of the model to the data set we collected from the internet. Section 5 contains a brief discussion and concluding comments.

2 Document-Entity-Topic Model

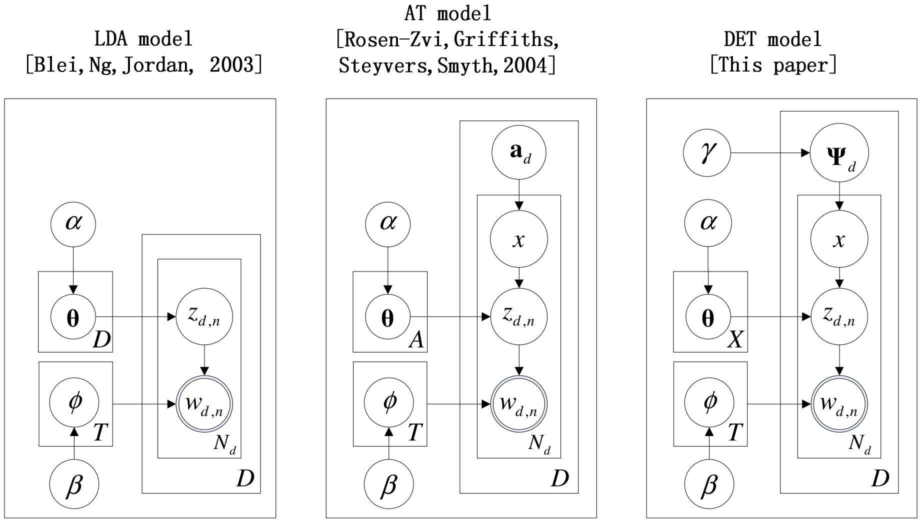
In this section we introduce the document-Entity-topic (DET) model. The DET model belongs to the family of generative models, in which each word w in a document is associated with two latent variables: an entity assignment x , and a topic assignment z .

2.1 Dirichlet Priori on Document-Entity Distribution

The entities in news may have different weights to be described by the words, for example, "a reporter covers that, Mr. A and B contact at a national conference and have an educational barging, trying to improve the intercommunion and friendship of both." We find that the relationship between A and B is closer than that between reporter and the two people. In order to discover the different weights for different entities, we use Dirichlet allocation as the prior distribution to describe the importance of each entity in a document, which is similar to LDA model by using Dirichlet allocation to describe the relationship between topics and the document.

The reason to choose the Dirichlet is that, firstly, it can reflect the characteristic of document-entity relation, a document has primary and minor entities, and the weights can be adjusted by the hyperparameter of Dirichlet. Secondly, the conjugate prior of multinomial distribution is Dirichlet allocation, so it can simplify the computation for the posterior distribution which has the same functional form as the prior.

Thus, we propose the Document-Entity-Topic (DET) model for mining the semantic description of entities and using the topical distribution to carry out the social networking tasks. The generative process of DET model for a document can be summarized as follows: firstly, an entity is chosen randomly from the distribution over entity-document; next, a topic label is sampled for each word from the distribution over topics associated with the entities of that word; finally, the words are sampled from the distribution over words associated with each topic. The plate representation[9] for all models are shown in figure 1.



LDA model
[Blei,Ng,Jordan, 2003]

AT model
[Rosen-Zvi,Griffiths,
Steyvers,Smyth,2004]

DET model
[This paper]

Fig. 1. Two related models and the DET model. In all models, each word *w* is generated from a topic-specific multinomial word distribution; however topics are sampled differently in each of the models. In LDA, a topic is sampled from a document-specific topic distribution which is sampled from a Dirichlet with hyperparameter. In the AT model, a topic is sampled from an author-specific multinomial distribution, and authors are sampled uniformly from the document's author set. In DET, Dirichlet prior has been introduced to the document-entity distribution, a topic is sampled from an entity-specific multinomial distribution, and entity assignment is sampled from the Dirichlet allocation of that document.

2.2 Generative Process of DET Model

In DET model, the generative process of generating a word is according to the probability distributions of firstly picking an entity followed by picking a topic.

a) For each document $d = 1, \ldots, D$ choose $\boldsymbol{\psi}_d \sim Dirichlet(\gamma)$;

For each entity $x = 1, \ldots, X$ choose $\boldsymbol{\theta}_x \sim Dirichlet(\alpha)$;

For each topic $t = 1, \ldots, T$ choose $\phi_t \sim Dirichlet(\beta)$.

b) For each document $d = 1, \ldots, D$

Given the vector of entities X_d, for each word w_i, out of the N_d words

Conditioned on $\mathbf{x}_d$ choose a persona $x_i \sim Dirichlet(\boldsymbol{\psi}_d)$;

Conditioned on x_i choose a topic $z_i \sim Dirichlet(\boldsymbol{\theta}_{x_i})$;

Conditioned on zi choose a word $w_i \sim Dirichlet(\phi_{z_i})$.

Under this generative process, each entity is drawn independently conditioned on $\boldsymbol{\psi}$; each topic is drawn independently conditioned $\boldsymbol{\Theta}$; and each word is drawn independently conditioned on $\boldsymbol{\Phi}$ and *z*. The probability of the corpus **w**, conditioned on $\boldsymbol{\psi}$, $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ is shown as equation (1):

$$P(\mathbf{w}|\boldsymbol{\Theta},\boldsymbol{\Phi},\boldsymbol{\Psi}) = \prod_{d=1}^{D} P(\mathbf{w}_d|\boldsymbol{\Theta},\boldsymbol{\Phi},\boldsymbol{\Psi}) \tag{1}$$

Summing over the latent variables *x* and *z*, we can obtain the probability of the words in each document $\mathbf{w}_d$ as equation (2):

$$\begin{aligned} P(\mathbf{w}_d \mid \boldsymbol{\Theta},\boldsymbol{\Phi},\boldsymbol{\Psi}) &= \prod_{i=1}^{N_d} P(\mathbf{w}_i \mid \boldsymbol{\Theta},\boldsymbol{\Phi},\boldsymbol{\Psi}) \\ &= \prod_{i=1}^{N_d}\sum_{x=1}^{X_d}\sum_{t=1}^{T} P(w_i, z_i = t, x_i = x \mid \boldsymbol{\Theta},\boldsymbol{\Phi},\boldsymbol{\Psi}) \\ &= \prod_{i=1}^{N_d}\sum_{x=1}^{X_d}\sum_{t=1}^{T} P(w_i \mid z_i = t,\boldsymbol{\Phi}) P(z_i = t \mid x_i = x,\boldsymbol{\Theta}) P(x_i = x \mid \boldsymbol{\Psi}) \\ &= \prod_{i=1}^{N_d}\sum_{x\in X_d} \psi_{x_d} \cdot \sum_{t=1}^{T} \theta_{xt} \cdot \phi_{w_i t} \end{aligned} \tag{2}$$

Factorizing in the third line of equation (2) uses the conditional independence assumptions of the model. The last line in the equations expresses the probability of the words *w* in terms of the parameter matrices $\boldsymbol{\psi}$, $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$. $P(x_i = x \mid \boldsymbol{\Psi})$ is the entity multinomial distribution $\boldsymbol{\psi}_d$ in $\boldsymbol{\Psi}$ which corresponds to document *d*, $P(z_i = t \mid x_i = x,\boldsymbol{\Theta})$ is the multinomial distribution $\boldsymbol{\theta}_x$ in $\boldsymbol{\Theta}$ that corresponds to entity *x*, and $P(w_i \mid z_i = t,\boldsymbol{\Phi})$ is the multinomial distribution ϕ_t in $\boldsymbol{\Phi}$ corresponding to topic *t*.

3 Learning the DET Model from Data

The DET model contains three continuous random variables $\boldsymbol{\Psi}$, $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$. The inference scheme used in this paper is based upon a Markov chain Monte Carlo (MCMC) algorithm or more specifically, Gibbs sampling. We estimate the posterior distribution $P(\boldsymbol{\Psi},\boldsymbol{\Theta},\boldsymbol{\Phi} \mid D^{train},\gamma,\alpha,\beta)$. The inference scheme is based upon the observation that

$$\begin{aligned} &P(\boldsymbol{\Psi},\boldsymbol{\Theta},\boldsymbol{\Phi} \mid D^{train},\gamma,\alpha,\beta) \\ &= \sum_{z,x} P(\boldsymbol{\Psi},\boldsymbol{\Theta},\boldsymbol{\Phi} \mid z,x,D^{train},\gamma,\alpha,\beta) P(z,x \mid D^{train},\gamma,\alpha,\beta) \end{aligned} \tag{3}$$

Where *z* is the topic variable and *x* is the entity assignment. This inference process involves two steps. Firstly, we use Gibbs sampling to obtain an empirical sample-based estimate of $P\left(z,x \mid D^{train},\gamma,\alpha,\beta\right)$. Second, we compute each specific sample corresponding to particular x and z using the conjugation trait between Dirichlet and multinomial distribution.

3.1 Gibbs Sampling

Gibbs sampling is a widely applicable Markov chain Monte Carlo algorithm which can be viewed as a special case of Metropolis Hastings algorithm. It often yields relatively simple algorithms for approximate inference in high-dimensional models such as topic models[9]. Here we wish to construct a Markov chain which converges to the posterior distribution over x and z in terms of D^{train},γ,α and β. Using Gibbs sampling we can generate a sample from $P\left(z,x \mid D^{train},\gamma,\alpha,\beta\right)$ by firstly sampling an entity assignment x_i and a topic assignment z_i for an individual word w_i conditioned on initialized assignments of entities and topics for all other words in the corpus. Secondly, repeating this process for each word. A single Gibbs sampling iteration consists of sequentially performing sampling of entity and topic assignments for each individual word in the corpus.

$P\left(z_i = t \mid \mathbf{z}_{-i},\mathbf{x},D^{train},\gamma,\alpha,\beta\right)$ and $P\left(x_i = x \mid \mathbf{x}_{-i},\mathbf{z},D^{train},\gamma,\alpha,\beta\right)$ can also be the Gibbs sampler. In this paper we use the blocked sampler where we sample x_i and z_i jointly. It can improve the mixing time of the sampler and the method also has been used similarly by Rosen-Zvi et al[5]. In Appendix, we derive the Gibbs sampler of document d and entity $x \in X_d$ as equation (4)

$$\begin{aligned} &P\left(x_i = x, z_i = t \mid w_i = w,\mathbf{x}_{-i},\mathbf{z}_{-i},\mathbf{w}_{-i},\gamma,\alpha,\beta\right) \\ &\propto \frac{C^{WT}_{wt,-i}+\beta}{\sum_{w'} C^{WT}_{w't,-i}+W\beta} \cdot \frac{C^{TX}_{tx,-i}+\alpha}{\sum_{t'} C^{TX}_{t'x,-i}+T\alpha} \cdot \frac{C^{XD}_{xd,-i}+\gamma}{\sum_{x'} C^{XD}_{x'd,-i}+X\gamma} \end{aligned} \tag{4}$$

Here C^{XD} represents the document-entity count matrix, where $C^{XD}_{xd,-i}$ is the number of words assigned to entity x for document d excluding word w_i. C^{TX} is the topic-entity count matrix, where $C^{TX}_{tx,-i}$ is the number of words assigned to topic t for entity x excluding the topic assignment to word w_i. C^{WT} is the number of words from the w^{th} entry in the vocabulary assigned to topic t excluding the topic assignment to word w_i. Finally, z_{-i}, x_{-i} , w_{-i} stand for the vector of topic assignments, vector of entity assignments and vector of word observations in corpus except for the i^{th} word, respectively.

3.2 The Posterior on $\mathbf{\Psi}$, $\mathbf{\Theta}$ and $\mathbf{\Phi}$

Given $\mathbf{z},\mathbf{x},D^{train},\gamma,\alpha$ and β , we can compute the posterior distributions on $\mathbf{\Psi}$, $\mathbf{\Theta}$ and $\mathbf{\Phi}$ directly. Using the fact that the Dirichlet is conjugate to the multinomial, we have

$$\phi_{w,t}=\frac{C^{WT}_{wt,-i}+\beta}{\sum_{w'}C^{WT}_{w't,-i}+V\beta},\theta_{t,x}=\frac{C^{TX}_{tx,-i}+\alpha}{\sum_{t'}C^{TX}_{t'x,-i}+T\alpha},\psi_{x,d}=\frac{C^{XD}_{xd,-i}+\gamma}{\sum_{x'}C^{XD}_{x'd,-i}+X\gamma} \tag{5}$$

These posteriors provide point estimates for $\mathbf{\Psi}$, $\mathbf{\Theta}$ and $\mathbf{\Phi}$. $\mathbf{\Psi}$ corresponds to the posterior predictive distribution for the documents and entities, it obeys the Dirichlet allocation other than uniform distribution, and can get the more valuable entities who effect the topics of the document more. $\mathbf{\Theta}$ corresponds to the posterior predictive distribution for the entities and topics, every entity has a vector of topics, it can tell us what topics the entity associates with and which entities are interested in the similar topics, so groups can be extracted from $\mathbf{\Theta}$. $\mathbf{\Phi}$ corresponds to the posterior predictive distribution for the topics and words, we can get the word description of topics.

4 Experiment Result

We train our DET model on the “Libya Event” dataset which is collected from Internet (http://www.ifeng.com). It contains $D=4165$ documents, $P=3784$ unique entities (most are person names), $N=782043$ tokens, and a vocabulary of $V=15812$ unique words. We preprocess the document set with tryout edition of ICTCLAS whose rights reserved by ictclas.org. All documents are written in Chinese, and we translate the results in English.

We run the Markov chain for a fixed number of 2000 iterations. Furthermore, we find that the sensitivity to hyperparameters is not very strong, so that we use the fixed symmetric Dirichlet distributions $\gamma=0.5, \alpha=0.1$, and $\beta=0.01$ in all our experiments. In the comparing experiment of AT model, the author set **a** are entities extracted from the documents.

4.1 Perplexity Comparison between AT and DET

Models for natural languages are often evaluated by perplexity as a measure of the goodness fit of models. The lower perplexity a model has, the better it predicts the unseen words. The perplexity of a previously unseen document *d* consisting of words $\mathbf{w}_d$ can be defined as equation (6) when the entities x_d are given:

$$Perplexity(\mathbf{w}_d)=\exp\left(-\frac{\log\left(p(\mathbf{w}_d \mid \mathbf{x}_d)\right)}{N_d}\right) \tag{6}$$

in which

$$p\left(\mathbf{w}_d \mid \mathbf{x}_d\right)=\prod_{i=1}^{N_d}\left(\sum_{x\in\mathbf{x}_d}\hat{\mathbf{\psi}}_x\cdot\sum_{t=1}^{T}\hat{\phi}_{w_it}\cdot\hat{\mathbf{\theta}}_{tx}\right)^{n_d^{(i)}} \tag{7}$$

Where $n_d^{(i)}$ is the number of times token *i* has been observed in document *d*. $\vec{\phi}_{w_i}$ can be determined by the training set, but $\mathbf{\theta}_x$ and $\mathbf{\psi}_d$ need to be derived by querying the model. Firstly, initializing the algorithm by randomly assigning topics and entities to words of the test documents, and then performing a number of loops through the Gibbs sampling update:

$$p\left(\tilde{z}_i = t, \tilde{x}_i = x \mid \tilde{w}_i = w, \mathbf{z}_{-i}, \mathbf{x}_{-i}, \mathbf{w}_{-i}; \mathrm{M}\right) \propto \varphi_{w\tilde{t}} \cdot \left(n_{x,-i}^{(\tilde{t})} + \alpha\right) \cdot \left(n_{d,-i}^{(\tilde{x})} + \gamma\right) \qquad (8)$$

Where $n_{x,-i}^{(\tilde{t})}$ is the number of topic *t* been assigned to persona *x*, and $n_{d,-i}^{(\tilde{x})}$ is the number of entity *x* been assigned to document *d*. Both of them exclude the topic and entity assignment of word w_i. We report the perplexities with different number of topics on "Libya Event" test data set with 109 documents, about 10% of the whole data set.

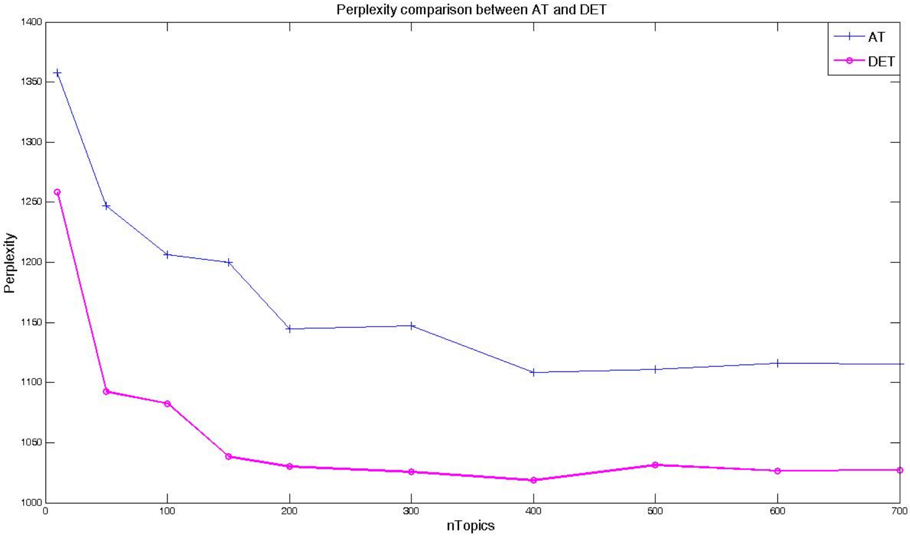


Fig. 2. Perplexity comparison of AT and DET on "Libya" data set. DET model has significantly better predictive power as AT over our document set. We can also find that the lowest perplexity obtained by DET is not achievable by AT with any topic number. It proves that DET can better adapt to the task of Semantic Social Network Analysis (SSNA), which discovers the topic-based relationship and group information of entities in documents.

4.2 Semantic Social Network Analysis with DET

Topics and Entities. We get the latent topics after applying the Gibbs sampling algorithm to DET model. We use the topic significance ranking method[10] to rank the topics and show two most important topics in table 1. In each topic we list the most likely words in the topic with their probability and below that the most likely entities and the topic names are named by authors.

During the experiment process, we have found that many topics own lots of same words with high probabilities, the reason we think is that all documents in "Libya Event" data set talk about one event (similar topics). We introduce in the idea of tf-idf algorithm to decide which words have high probabilities. The probability of word w belonging to topic k depends on both of the DET result, i.e., $\phi_{k,w}$ and the *tf_idf* value, which ranges from 0 to 1 with standardization. So the final probability of a word belonging to a topic is $\phi'_{k,w} = \delta \cdot \phi_{k,w} + (1-\delta) \cdot tf\_idf_{k,w}, 0 < \delta < 1$.

The probability of entity x belonging to topic k is not only decided by $\theta_{x,k}$, but also decided by the number of entity x appearing in documents. If x appears in document d, the number adds 1, and the appearing frequency is $df_x = |\{x \in d\}|/|D|$. So the probability of entity x with topic k is $\theta'_{x,k} = df_x \cdot \theta_{x,k}$.

Table 1. Two topics with highest probabilities from a 100-topic DET running with "Libya Event" data. In each topic we list the most likely words in lowercase with their probabilities, and below that the most likely entities in uppercase with initial.

Topic89: Conflicts of government and opposition in Libya		Topic31: National transition committee comes into existence	
the opposition	0.751071	committee	0.313636
demos	0.098968	transition	0.278701
fremdness	0.018027	nation	0.213317
relation	0.015836	admit	0.046756
find out	0.013272	chairman	0.033091
reason	0.011508	come into existence	0.024036
in the past	0.008329	spokesman	0.020482
hours	0.007549	intraday	0.013421
with responsibility for	0.006162	leaguer	0.013068
encounter	0.005506	promise	0.006441
Qaddafi	0.060839	Abdul-Jelil	0.041501
Bangh acirc	0.043085	Bangh acirc	0.026637
Qatar	0.030726	Italy	0.025412
Reuters	0.027212	National Transition Committee	0.025164
Italy	0.020556	Qatar	0.023441
Russia	0.014663	Bani Walid	0.019576
Abdul-Jelil	0.014444	Abdul-Jelil	0.016731
Egypt	0.013855	Beijing	0.016373
Associated Press	0.008018	Paris	0.015959
Muhammad	0.007668	London	0.015528

Entity Significance Ranking. We suppose that if the topic distribution of an entity is much related to that of the document, the entity is significant to this document. Usually, KL divergence is used to measure the similarity between the entity and document. We show the KL divergences, probabilities and frequencies of all entities in two documents for particular information in table 2.

Table 2. KL divergences, probabilities and frequencies of all entities in two documents for particular information

The National transition committee encounters plaster in Surt			
entity	**KL divergence**	**probability**	**frequency**
Misracirctah	0.205826	0.181452	1
Abdul-Jelil	0.266745	0.181452	1
Qaddafi	0.485460	0.149194	10
Saif-Nasser	0.498184	0.125	1
New York	0.435166	0.084677	2
Niger	0.411440	0.060484	1
Surt	0.598851	0.044355	3
Bani Walid	0.494996	0.03629	1
Tripoli	0.635240	0.03629	1
Jerusalem	0.624283	0.028226	1
UN’s high conference of Libya appeals to picking up the reconstruction			
entity	**KL divergence**	**probability**	**frequency**
Abdul-Jelil	0.018252	0.444015	1
Wei Wei	0.591291	0.374517	1
Libya	0.642270	0.104247	16
UN	0.662659	0.042471	6
New York	0.671120	0.019305	1
Gu Zhengqiu	0.669310	0.003861	1
XinHua Net	0.689017	0.003861	1
UNSC	0.652071	0.003861	1
Ban ki-moon	0.666539	0.003861	1

In most instances, if an entity which has a lower KL divergence with the document, the probability it belongs to that document will be higher, and the frequency is not a key factor to influence the belonging probability. In order to compare the entity ranking performances between AT and DET model on the whole data, we further adopt the weighted KL divergence which is defined as equations (9) and (10):

$$wKL\_AT = \frac{1}{D}\cdot\frac{1}{\mathbf{a}_d}\sum_{d=1}^{D}\sum_{a=1}^{\mathbf{a}_d} KL\left(\theta_{x,t} \parallel \eta_{d,t}\right) \tag{9}$$

$$wKL\_DET = \frac{1}{D} \cdot \sum_{d=1}^{D} \sum_{x=1}^{X_d} \left(\psi_{d,x} \cdot KL\left(\theta_{x,t} \, \| \, \eta_{d,t} \right) \right) \tag{10}$$

The smaller the weighted KL value is, the more similarity entities and documents own. In figure 3, we have shown the values with different topic numbers.

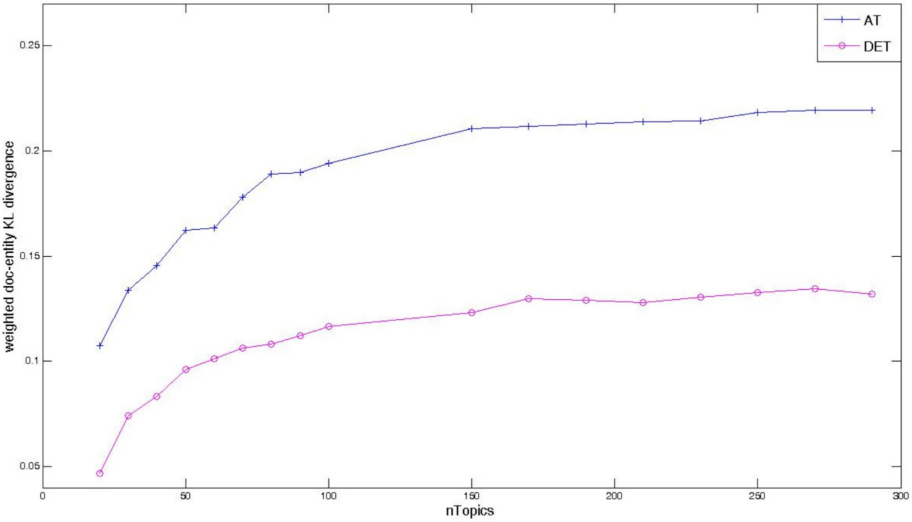


Fig. 3. The weighted KL divergences of AT and DET model with different topic numbers. The values of DPT model are lower than AT model. It means that the more important entities (with lower KL divergence to document which they appear in) have higher probabilities belong to the document and contributes more to the topic generativity.

5 Conclusions

We have presented the Document-Entity-Topic model, a probabilistic model for exploring the interactions of words, topics and entities within documents. It applies the probabilistic model to the social network analysis based on latent topics. In order to avoid the side effects of noisy entities and find out the entities which mainly affect the topics, we have introduced in the Dirichlet allocation for document-entity distribution other than uniform allocation. The model can be applied to discovering topics conditioned on entities, clustering to find semantic social groups, and ranking the significance of entities in a document.

However, while there is no entity in a document, the topics of that document can not be modeled. When such lack-of-entity documents arrive at a certain amount, the topic modeling of the corpus will be affected. Consequently, we try to improve the model for the application when there are many documents lacking of entities.

References

1. Blei, D.: Introduction to Probabilistic Topic Models. Communications of the ACM (2011)
2. Steyvers, M., Griffiths, T.: Probabilistic Topic Models. Handbook of Latent Semantic Analysis 427 (2007)
3. Blei, D., Carin, L., Dunson, D.: Topic Models. IEEE Signal Processing Magazine 27(6), 55–65 (2010)
4. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet Allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
5. Rosen-Zvi, M., Chemudugunta, C., Griffiths, T., Smyth, P., Steyvers, M.: Learning Author-Topic Models from Text Corpora. ACM Transactions on Information Systems (TOIS) 28(1), 1–38 (2010)
6. Rosen-Zvi, M., Griffiths, T., Steyvers, M., Smyth, P.: The Author-Topic Model for Authors and Documents, pp. 478–494. AUAI Press (2004)
7. Shiozaki, H., Eguchi, K., Ohkawa, T.: Entity Network Prediction Using Multitype Topic Models. Springer (2008)
8. Newman, D., Chemudugunta, C., Smyth, P.: Statistical Entity-Topic Models, pp. 680–686 (2006)
9. Bishop, C.: Pattern Recognition and Machine Learning. Springer, New York (2006)
10. AlSumait, L., Barbará, D., Gentle, J., Domeniconi, C.: Topic Significance Ranking of LDA Generative Models. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009. LNCS, vol. 5781, pp. 67–82. Springer, Heidelberg (2009)

Appendix

We need to derive $P\left(x_i = x, z_i = t \mid w_i = w, \mathbf{x}_{-i}, \mathbf{z}_{-i}, \mathbf{w}_{-i}, \gamma, \alpha, \beta\right)$, the conditional distribution for word w_i given all other words' topic and entity assignments z_{-i} and x_{-i} to give out the Gibbs sampling procedure for DET model. We begin with the joint probability of the whole documents corpora. Here we can make use of conjugate priors to simplify the integrals.

$$
\begin{aligned}
& P\left(\mathbf{x},\mathbf{z},\mathbf{w} \mid \gamma,\alpha,\beta,X\right) \\
&= \int\int\int \prod_{d=1}^{D} p(\boldsymbol{\psi}_d \mid \gamma) \prod_{x=1}^{X} p(\boldsymbol{\theta}_x \mid \alpha) \prod_{t=1}^{T} p(\phi_t \mid \beta) \prod_{i=1}^{N_d} p(x_{di} \mid \boldsymbol{\psi}_d) p(z_{di} \mid \boldsymbol{\theta}_{dx_{di}}) P(w_{di} \mid \phi_{z_{di}}) d\boldsymbol{\Phi} d\boldsymbol{\Theta} d\boldsymbol{\Psi} \\
&= \int \prod_{d=1}^{D} \left(\frac{\Gamma(\sum_{d=1}^{D} \gamma_x)}{\prod_{x=1}^{X} \Gamma(\gamma_x)} \prod_{x=1}^{X} \boldsymbol{\psi}_{dx}^{\gamma_d - 1} \right) \prod_{d=1}^{D} \prod_{x=1}^{X} \boldsymbol{\psi}_{dx}^{n_{dx}} d\boldsymbol{\Psi} \\
&\quad \times \int \prod_{x=1}^{X} \left(\frac{\Gamma(\sum_{t=1}^{T} \alpha_t)}{\prod_{t=1}^{T} \Gamma(\alpha_t)} \prod_{t=1}^{T} \boldsymbol{\theta}_{xt}^{\alpha_t - 1} \right) \prod_{x=1}^{X} \prod_{t=1}^{T} \boldsymbol{\theta}_{xt}^{n_{xt}} d\boldsymbol{\Theta} \times \prod_{t=1}^{T} \left(\frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\prod_{v=1}^{V} \Gamma(\beta_v)} \prod_{v=1}^{V} \phi_{tv}^{\beta_v - 1} \right) \prod_{t=1}^{T} \prod_{v=1}^{V} \phi_{tv}^{n_{tv}} d\boldsymbol{\Phi} \\
&\propto \prod_{d=1}^{D} \int \prod_{x=1}^{X} \boldsymbol{\psi}_{dx}^{\gamma_d + n_{dx} - 1} d\boldsymbol{\psi}_d \times \prod_{x=1}^{X} \int \prod_{t=1}^{T} \boldsymbol{\theta}_{xt}^{\alpha_t + n_{xt} - 1} d\boldsymbol{\theta}_x \times \prod_{t=1}^{T} \int \prod_{v=1}^{V} \phi_{tv}^{\beta_v + n_{tv} - 1} d\phi_t \\
&\propto \prod_{d=1}^{D} \frac{\prod_{x=1}^{X} \Gamma(\gamma_x + n_{dx})}{\Gamma\left(\sum_{x=1}^{X} (\gamma_x + n_{dx})\right)} \times \prod_{x=1}^{X} \frac{\prod_{t=1}^{T} \Gamma(\alpha_t + n_{xt})}{\Gamma\left(\sum_{t=1}^{T} (\alpha_t + n_{xt})\right)} \times \prod_{t=1}^{T} \frac{\prod_{v=1}^{V} \Gamma(\beta_v + n_{tv})}{\Gamma\left(\sum_{v=1}^{V} (\beta_v + n_{tv})\right)}
\end{aligned}
$$

Where n_{dx} is the number of tokens assigned to persona x and document d, n_{xt} is the number of tokens assigned to topic t and persona x, n_{tv} is the number of tokens of word w assigned to topic t. Using the chain rule, we can obtain the conditional probability conveniently. We define w_{-di} as all word tokens except the token w_{di} :

$$
\begin{aligned}
& P(x_{di}, z_{di} \mid \mathbf{x}_{-di}, \mathbf{z}_{-di}, \mathbf{w}, \alpha, \beta, \gamma, \mathbf{X}) \\
&= \frac{P(x_{di}, z_{di}, w_{di} \mid \mathbf{x}_{-di}, \mathbf{z}_{-di}, \mathbf{w}_{-di}, \alpha, \beta, \gamma, \mathbf{X})}{P(w_{di} \mid \mathbf{x}_{-di}, \mathbf{z}_{-di}, \mathbf{w}_{-di}, \alpha, \beta, \gamma, \mathbf{X})} \\
&\propto \frac{P(\mathbf{x},\mathbf{z},\mathbf{w} \mid \alpha, \beta, \gamma, \mathbf{X})}{P(\mathbf{x}_{-di}, \mathbf{z}_{-di}, \mathbf{w}_{-di} \mid \alpha, \beta, \gamma, \mathbf{X})} \\
&\propto \frac{\gamma_{x_{di}} + n_{dx_{di}} - 1}{\sum_{x=1}^{X} (\gamma_x + n_{dx}) - 1} \frac{\alpha_{z_{di}} + n_{x_{di} z_{di}} - 1}{\sum_{z=1}^{T} (\alpha_z + n_{x_{di} z}) - 1} \frac{\beta_{w_{di}} + n_{z_{di} w_{di}} - 1}{\sum_{v=1}^{V} (\beta_v + n_{z_{di} v}) - 1}
\end{aligned}
$$

Visualizing Clusters in Parallel Coordinates for Visual Knowledge Discovery

Yang Xiang[1], David Fuhry[2], Ruoming Jin[3], Ye Zhao[3], and Kun Huang[1]

[1] Department of Biomedical Informatics, The Ohio State University,
Columbus, OH 43210, USA
{yang.xiang,kun.huang}@osumc.edu

[2] Department of Computer Science and Engineering, The Ohio State University,
Columbus, OH 43210, USA
fuhry@cse.ohio-state.edu

[3] Department of Computer Science, Kent State University,
Kent, OH 44242, USA
{jin,zhao}@cs.kent.edu

Abstract. Parallel coordinates is frequently used to visualize multi-dimensional data. In this paper, we are interested in how to effectively visualize clusters of multi-dimensional data in parallel coordinates for the purpose of facilitating knowledge discovery. In particular, we would like to efficiently find a good order of coordinates for different emphases on visual knowledge discovery. To solve this problem, we link it to the metric-space Hamiltonian path problem by defining the cost between every pair of coordinates as the number of inter-cluster or intra-cluster crossings. This definition connects to various efficient solutions and leads to very fast algorithms. In addition, to better observe cluster interactions, we also propose to shape clusters smoothly by an energy reduction model which provides both macro and micro view of clusters.

Keywords: Multi-dimensional Data Visualization, Parallel Coordinates, Cluster, knowledge discovery, Graph Theory, Metric Space, Metric Hamiltonian path problem.

1 Introduction

Today the infusion of data from every facet of our society, through documenting, sensing, digitalizing and computing, challenges scientists, analysts and users with its typical massive size and high dimension. Data mining and visualization are two important areas in analyzing and understanding the data. The role of data mining is to discover hidden patterns of the data. In particular, various clustering techniques (see [19] for a review) have been proposed to reveal the structures of these data and support exploratory data analysis. By contrast, the role of visualization is to present the data in a clear and understandable manner for people. Many visualization techniques have been developed to facilitate exploratory analysis and analytical reasoning through the use of (interactive) visual interfaces.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 505–516, 2012.

However, despite these efforts, current research is far from perfect in integrating these two endeavors in a close and uniform fashion. Given the discovered cluster structures from the data, how can we visualize them and provide users better insight of the data? How can visualization techniques help reveal and expose underlying structures of the data? Those research questions are clearly very critical for us to meet the challenges of the "data explosion". In this paper, we address those questions by developing a novel visualization model for visualizing discovered clusters in large and multivariate datasets. Our goal is to efficiently provide users different views of discovered clusters as well as preserve the details of these clusters to the maximal extent possible. Among many visualization techniques for multidimensional data [33], parallel coordinates is one of the most elegant yet simple tools, and we select it as the visualization platform for our proposed algorithms.

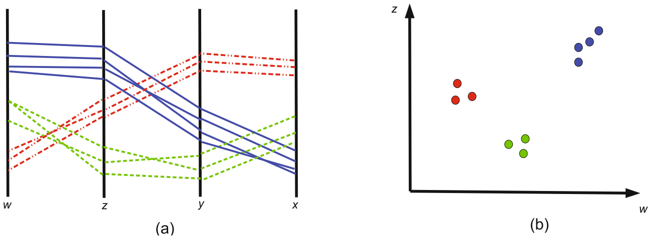


Fig. 1. (a) Data visualization with parallel coordinates w, x, y, z; (b) Data projection on wz plane

Parallel coordinates, which transforms multivariate data into 2D polylines (or 'lines' for short), has been widely used in many information visualization applications [18,28] as well as data mining [35]. Figure 1(a) is an example of visualizing 4-dimensional data by parallel coordinates. Wegman [30] shows Parallel Coordinates can be used to effectively reveal data correlation as well as cluster interaction. For instance, Figure 1(b) is the projection of three clusters in Figure 1(a) on the wz plane. Readers can easily observe what it implies for two clusters that are generally crossing over each other between two coordinates (e.g. wz coordinates in Figure 1), or generally parallel to each other. Such observation leads to important knowledge discovery. Using an example in [30] as an illustration, let us imagine we are comparing one group of heavy cars with another group of light cars, on weight, displacement, mileage, gear ratio, and price. A visualization of these two clusters on gear ratio and weight would show these two clusters generally cross each other, which implies that heavy cars would tend to have a large engine but a lower gear ratio, while light cars are just the reverse.

As there is a factorial number of ways to order the parallel coordinates for visualizing different aspects of the data, a major challenge arises: How to efficiently determine a good order of coordinates (i.e., columns or dimensions) for a specific knowledge discovery purpose? Such order is traditionally pre-determined, and made flexible in some systems by allowing user adjustment. For data sets with many dimensions, this will impose unexpected challenge to end users, while they may have inadequate knowledge and experiences. Moreover, data clusters from aggregation and abstraction are even harder to be illustrated along multiple coordinates, together with many polylines.

To address these challenges, in this paper, we visualize clusters in parallel coordinates for visual knowledge discovery using a novel dimension ordering approach which is further refined by an energy reduction model.

2 Related Works

Parallel Coordinates for Clusters. Wegman [30] promotes parallel coordinates visualization in the aspects of geometry, statistics and graphics, which has been widely applied in information visualization [28,25]. For visualizing clustered data sets, many approaches have been conducted using parallel coordinates [20,23,3]. In particular, instead of visualizing each individual data item as a polyline, each cluster pattern is visualized as a fuzzy stripe [13,20,3]. Fua et al. [9] visualize clusters by variable-width opacity bands, faded from a dense middle to transparent edges. Such visualization focuses on the global pattern of clusters, but the general shape of a cluster might be adversely affected by a small number of outliers inside a cluster. In comparison, instead of displaying a shape profile of individual clusters, our method seeks to keep the line structures while highlighting clusters and their relationships, by seeking good orders of coordinates as well as shaping them smoothly by a quadratic energy reduction model which extends the linear system proposed in [36].

Dimension Ordering. The dimension ordering and permutation problem is naturally associated with parallel coordinate visualization. It is discussed in the early paper by Wegman [30] and the subsequent work by Hurley and Oldford [17,16]. In [30],Wegman points out the problem and gives a basic solution on how to enumerate the minimum number permutations such that every pair of coordinates can be visualized in at least one of the permutations. However, it is rather inefficient to display parallel coordinates corresponding to all these permutations. The grand tour animates a static display in order to examine the data from continuous different views [29,32,31]. The method is effective by seeking solution to temporal exploration for computational complex tasks such as manifesting outliers and clusters. Ankerst et al. [1] propose to rearrange dimensions such that dimensions showing a similar behavior are positioned next to each other. Peng et al. [27] try to find a dimension ordering that can minimize the “clutter measure”, which is defined as the ratio of outliers to total data points. Since permutation related problems are mostly NP-hard, the existing work [1,27,34] primarily relies on heuristic algorithms to get a quick solution.

Ellis and Dix [8] use line crossings to reduce clutter. Dasgupta and Kosara [7] recently use number of crossings as an indication of clutter between two adjacent coordinates, and apply a simple Branch-and-Bound optimization for dimension ordering. Hurley [15] uses crossings to study the correlation between two dimensions of a dataset as well as reduce clutters. Different from them, we define the crossing as an order change between a pair of inter-cluster items (or intra-cluster items, depending on the visualization focus) on two adjacent coordinates. Our definitions lead to an effective and efficient solution to study cluster interactions on parallel coordinates for visual knowledge discovery.

3 Dimension Ordering for Knowledge Discovery

Compared with the method of projecting data into a two dimensional plane for analysis (e.g. Figure 1(b)), an n-dimensional dataset visualized by n parallel coordinates is more efficient for data analysis as it displays data in $n-1$ pairs of dimensions at one time. It is obvious that different permutations of the n dimensions show different aspects of the dataset. For datasets with discovered clusters, different permutations give different views on the relations of those clusters. As shown in Figure 1, the overall crossing between red and green clusters on wz coordinates implies they are generally separable by a $z = w + c'$ line on the zw plane, while the overall non-crossing between blue cluster and red (or green) clusters on wz coordinates implies they are generally separable by a $z = -w + c''$ line on the zw plane. More importantly, cluster interactions often connect to important knowledge discovery, as the large car and small car example in Section 1 tells us. With today's data explosion in many applications people often have very limited time on viewing a dataset and would like to see the most informative aspect at the first look. To fit these applications we do not use coordinate permutation strategies in [30,17,16] (which generate many views of a dataset) to visualize cluster interactions. Instead, we ask the following question.

> *Is it possible to quickly provide users a suggestive order of coordinates to view cluster relationships for some given preference?*

In statistics, people use data correlation (e.g. Pearson correlation) to describe the relation between two vectors. However, there is no widely adopted similar measurement for the relation between two clusters to our knowledge. Thus, analogous to data correlation, we use inter-cluster relation to describe the interaction between two clusters. We consider two clusters are clear positively-related if they are generally in parallel or have few crossings, and two clusters are clear negatively-related if they are generally crossing each other. The quantitative measurement of the interaction between two clusters is the number of inter-cluster crossings between them. This measurement links to knowledge discovery on cluster interactions.

Similarly, one can also define intra-cluster crossings, which reveals relations among data within a cluster. A coordinate order that minimizes intra-cluster crossings also has significant meanings in knowledge discovery. It reduces visual clutter caused by data interactions within a cluster, and thus is more likely to manifest inter-cluster relations to users. We provide the definitions in the following.

3.1 Inter-cluster and Intra-cluster Crossings

An *inter-cluster crossing* is defined as an order change between two items from two different clusters on two coordinates. For example, for two items $i \in Cluster_\alpha$ and $j \in Cluster_\beta$ on two coordinates x and y, if $x_i \prec x_j$ and $y_i \succ y_j$, then we say an inter-cluster crossing exists between item i and j on the xy-dimension. Similarly, one can define an *intra-cluster crossing* as an order change between two items from the same cluster on two coordinates.

Assume σ_x and σ_y are the order of data on the x-coordinate and the y-coordinate, respectively. Our definitions can be formalized as follows:

Definition 1. *The number of inter-cluster crossings between $Cluster_\alpha$ and $Cluster_\beta$ on the coordinates x and y is $|C_{(\alpha,\beta)}|$ where $C_{(\alpha,\beta)} = \{(i,j)|\sigma_x(i) \prec \sigma_x(j)$ and $\sigma_y(j) \prec \sigma_y(i)$ and $i \in Cluster_\alpha, j \in Cluster_\beta\}$. The number of intra-cluster crossings among $Cluster_\alpha$ on the coordinates x and y is $|C_\alpha|$ where $C_\alpha = \{(i,j)|\sigma_x(i) \prec \sigma_x(j)$ and $\sigma_y(j) \prec \sigma_y(i)$ and $i, j \in Cluster_\alpha\}$.*

Definition 2. *The number of total inter-cluster crossings on the coordinates x and y is $|A|$ where $A = \{(i,j)|\sigma_x(i) \prec \sigma_x(j)$ and $\sigma_y(j) \prec \sigma_y(i)$ and i, j belong to different clusters$\}$. The number of total intra-cluster crossings on the coordinates x and y is $|B|$ where*
$B = \{(i,j)|\sigma_x(i) \prec \sigma_x(j)$ and $\sigma_y(j) \prec \sigma_y(i)$ and i, j belong to the same cluster$\}$.

According to Definitions 1 and 2, we can calculate the four types of crossings on a pair of coordinates in $O(n^2)$ time, where n is the number of data items (i.e. lines in the parallel coordinates). It is interesting to observe that the definition of intra-cluster crossing among one given cluster on a pair of coordinates (Second part of Definition 1) corresponds to the Kendall's Tau coefficient [21,26] in statistics, and there is a $O(n \log n)$ algorithm [22] for calculating it. Although the inter-cluster crossings do not correspond to the Kendall's Tau coefficient, it is not difficult to design a $O(n \log n)$ algorithm to calculate each type of crossings defined in Definitions 1 and 2 (assuming the number of clusters is a constant). We omit further details due to the space limit.

3.2 Optimization with Hamiltonian Path

After we get the number of crossings between every pair of coordinates, we need to find an order of coordinates such that the number of crossings is minimized or maximized for different knowledge discovery purposes. This problem can be converted to the problem of finding a minimum (or maximum) weighted Hamiltonian path [10] in a complete graph, by turning each coordinate into a vertex, adding an edge between every two vertices, and setting the edge weight to be the number of crossings between the two corresponding coordinates. It is quite obvious that the minimum or maximum weighted Hamiltonian path problem for complete graphs is NP-hard, as it is easy to reduce the Hamiltonian path problem for an unweighted graph, which is NP-complete, to this problem.

Exact Solution. An exact solution for the minimum (or maximum) weighted Hamiltonian path problem exhaustively tries all the permutations of vertices. The complexity is $O(n!)$ and the method becomes intractable when n is slightly larger. However, in parallel coordinates visualization, it is not uncommon to see a dataset with 10 or less coordinates. For these applications, the exhaustive search algorithm is still one of the most simple and effective solutions. Ideas in various branch and bound approaches for the Traveling Salesman Problem (TSP for short) can be used to speed up the exhaustive search algorithm for the minimum (or maximum) weighted Hamiltonian path problem. Interested readers may refer to the TSP survey paper [24] for details.

Metric Space and Approximation Solutions. Since the exact solutions cannot easily handle high-dimensional data, we seek fast approximate solutions when the number of coordinates is large. As nice approximate algorithms for minimum or maximum metric-TSPs exist (see solutions in [5] for minimum metric-TSP, [12] for maximum Metric-TSP, [4] for minimum metric-TSP with a prescribed order of vertices), we are wondering if our problems are metric Hamiltonian path problems. If they are, can we have similar approximate algorithms? Fortunately, we have a positive answer as stated in Lemma 1 (proof omitted due to space limit) which extends the well-known fact that Kendall tau distance (corresponds to intra-cluster crossings) is a metric :

Lemma 1. *The graph G, constructed by converting each coordinate to a vertex and setting the weight of each edge between two vertices to be the number of inter-cluster crossings between the two corresponding coordinates, either within two specific clusters or among all clusters, forms a metric space, in which edge weights follow the triangle inequality.*

Thus, it is not difficult to show that, if a graph G, with n vertices forms a metric space (regardless whether there exists a prescribed order of some vertices), a k-approximation solution for the minimum (or maximum) traveling salesman problem implies $2k$-approximation solutions for minimizing (or maximizing) inter-cluster (or intra-cluster) crossings.

In some special cases, it is possible to achieve even better approximation ratio. For example, Hoogeveen [14] shows that Christofides' 1.5 Approximation algorithm [5] of minimum metric TSP can be modified for minimum metric Hamiltonian path problem with the same approximation ratio, but the time complexity of this algorithm or its modified version, though polynomial, is much larger than linear. To achieve an even faster running speed for minimizing inter-cluster (or intra-cluster) crossings, we implemented a linear 2-approximation minimum metric Hamiltonian algorithm modified from the well-known *linear* 2-approximation algorithm for the minimum metric-TSP [6].

3.3 Empirical Study on Real Datasets

In this subsection, we report our empirical results on data extracted from the UC Irvine Machine Learning Repository[1], which has been widely used as a primary

[1] http://archive.ics.uci.edu/ml/

Table 1. Dataset characteristics and number of crossing changes

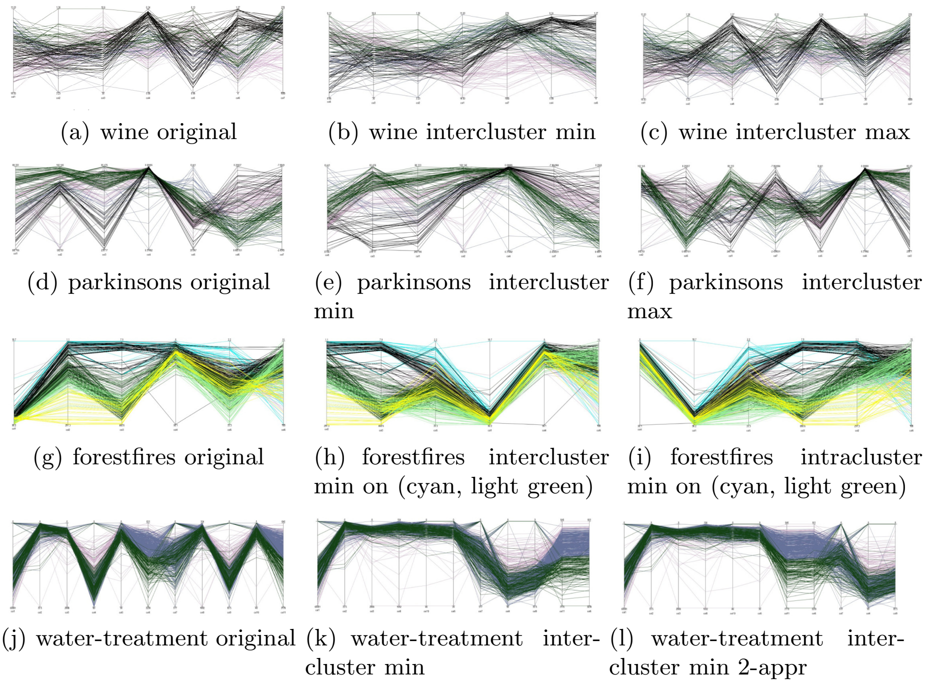
	dataset characteristics			number of crossing changes		
Dataset	Records	Columns	Clusters	inter min	inter max	intra min
eighthr	2533	12	2	-24.3%	+50.0%	-15.1%
forestfires	517	6	6	-29.6%	+24.7%	-21.9%
parkinsons	194	7	4	-42.0%	+40.8%	-26.3%
pima-indians	767	7	10	-15.2%	+20.6%	-15.4%
water-treatment	526	11	3	-41.9%	+13.6%	-37.0%
wdbc	568	5	4	-14.3%	+20.2%	-10.6%
wine	177	7	4	-46.8%	+13.4%	-11.0%

source of machine learning and data mining datasets. The basic characteristics of the datasets to be studied, are listed in Table 1. For our experiments, we chose the well-known K-means algorithm [11] to cluster the data items into exclusive clusters. We implemented the visualization program in JavaScript (web-based). For this study, we tested our visualization implementation in Firefox 3.6.12 on a mainstream desktop PC with an Intel Core i5 2.67GHz CPU and 8 GB of memory.

In our empirical study, we are primarily interested in observing the effects of proposed inter-cluster and intra-cluster ordering for visual knowledge discovery. Maximizing intra-cluster crossings does not clearly connect to the study of cluster interactions thus we omit it for the conciseness of the paper.

Table 1 reports the detailed changes of inter-cluster crossings after minimization and maximization, and intra-cluster crossings after minimization, for different datasets. Although crossing changes are substantial, it is more interesting to see what are the changes on the visualization results? A set of representative results are as shown in Figure 2.

Minimizing and Maximizing Total Inter-cluster Crossings: Figure 2 (b) and (c) shows the visualization results for minimizing total inter-cluster crossings and maximizing total inter-cluster crossings, respectively, for dataset "wine". In the original order, i.e., Figure 2 (a), we can observe clusters are generally negatively related between col 3 and col 4, between col 4 and col 5, between col 5 and col 6, between col 6 and col 7. Quite impressively, clusters show much more positive relations in the adjacent coordinates in Figure 2 (b). In contrast, clusters show even more negative relations in Figure 2 (c). These results generally meet our expectation for the effects of minimizing and maximizing the total inter-cluster crossings. Interestingly, we can observe the last two columns (col 3 and col 7) in Figure 2 (c) contain a couple of strongly negatively-related cluster pairs which are not revealed by Figure 2 (a) on its original order. By checking the original data, we found col 3 corresponding to "alkalinity of ash", while col 7 corresponding to "proline". This helps explain the negative relations between clusters as alkalinity is the ability of a solution to neutralize acids, while the proline is an α-amino acid. A high in alkalinity is more likely to result in low α-amino acid.



(a) wine original
(b) wine intercluster min
(c) wine intercluster max
(d) parkinsons original
(e) parkinsons intercluster min
(f) parkinsons intercluster max
(g) forestfires original
(h) forestfires intercluster min on (cyan, light green)
(i) forestfires intracluster min on (cyan, light green)
(j) water-treatment original
(k) water-treatment intercluster min
(l) water-treatment intercluster min 2-appr

Fig. 2. Visualization results (colors shown in the web version of this paper)

Similarly, Figure 2 (e) and (f) shows the visualization results for minimizing total inter-cluster crossings and maximizing total inter-cluster crossings, respectively, for dataset "parkinsons", in which each col represents a measurement for "parkinsons". After our visualization, it is easy for health care providers to spot measurements that are strongly positively-related in Figure 2 (e), and measurements that are strongly negatively-related in Figure 2 (f).

Minimizing Inter-cluster and Intra-cluster Crossings on a Pair of Clusters:
We would like to see the difference between minimizing inter-cluster crossings and intra-cluster crossings. To ease our observation, we focus on only two clusters, cyan and light green, in Figure 2 (g), which shows the dataset "forestfires" in its original order. Figure 2 (h) and (i) show the visualization results corresponding to minimizing inter-cluster crossings and intra-cluster crossings, respectively. In Figure 2 (h) we can observe that the cyan cluster and the light green cluster are generally positively-related in all adjacent columns. This is understandable as the visualization goal is to minimize the inter-crossings between them. However, Figure 2 (i) shows a strongly negative-relation between them on the last two columns (col 2 and col 6). This is because the goal of minimizing intra-cluster crossings does not care about the relations between the cyan cluster and light green cluster. Rather, it tries to reduce crossing within the two clusters so as

to reduce visual clutter and provide a better chance to observe the relations, regardless of positive or negative, between the two clusters.

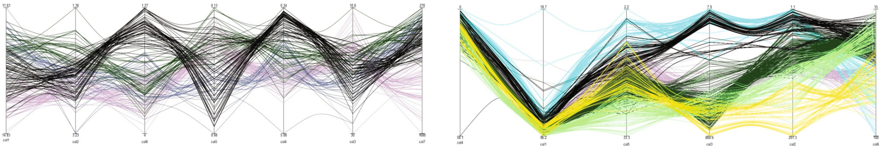
By checking the original data, we found col 2 corresponding to DMC and col 6 corresponding to RH. DMC is an indication (the larger the more likely) of the depth that fire will burn in moderate duff layers and medium size woody material, while RH is relative humidity. Thus, we understand the discovered result in Figure 2 (i) that clusters tend to be negatively-related between DMC and RH.

Minimizing Inter-cluster Crossings by the 2-Approximation Algorithm: In all the tested datasets, the exact algorithm finishes in no more than 100 milliseconds except for the datasets "eighthr" and "water-treatment". It takes about 2 minutes to exactly order "eighthr" (12 columns), and about 15 seconds to exactly order "water-treatment" (11 columns). This poses a concern on using exact algorithms for ordering datasets with more than 10 columns, and justify the importance of approximation algorithms for ordering large datasets. In the following we empirically study the effect of the popular 2-approximation algorithm (discussed at the end of Section 3.2) on our visualization scheme. In order to get a better ordering through the 2-approximation algorithm, we try DFS search from each vertex and find a lowest-cost result among all the 2-approximation results. Even with multi-DFS search, the ordering time is still lightning fast. For all datasets, including "eighthr" and "water-treatment", the multi-DFS search finishes within a couple of milliseconds. This makes our visualization schemes work for large datasets.

Figure 2 (l) shows the visualization result of minimizing inter-cluster crossings by the 2-approximation algorithm. Compared to the visualization result by the exact algorithm as in Figure 2 (k), it is hard to tell the actual difference between the two algorithms in revealing the positive relations among clusters. Detailed data may explain this: The numbers of inter-cluster crossings minimized by the 2-approximation algorithm are -23.0%,-29.6%,-42.0%,-8.4%,-41.6%,-14.3%,-46.8%, respectively, for the datasets in Table 1 (from top to bottom). Thus we can see there is very little performance degradation (in some datasets there is no difference) with the 2-approximation algorithm but very significant speed-up (linear vs factorial, in terms of complexity).

4 Shaping Clusters against Visual Clutters by an Energy Reduction Model

For some figures (e.g., Figure 2(c) and (i)) in the previous section, inter-cluster crossings are hard to discern even after ordering the coordinates for minimizing intra-cluster crossings. This is because a substantial amount of lines from a large-scale data set are typically entangled together in the limited space and resolution of display devices, confounding their belongings to different clusters. Consequently, the pattern and knowledge discovery of clustered data is hindered and the usage of parallel coordinates is limited. A handful of works [3,2,20] display the silhouette shape of individual clusters while the lines are intentionally brushed out. The shape of a cluster is sensitive to a few outliers, and as a result, the visualization of cluster relations is not satisfying. This scenario can be further deteriorated, when the lines of a given cluster are even more sparsely distributed.



(a) enhancement for Figure 2(c)

(b) enhancement for Figure 2(i)

Fig. 3. visualization enhancement (colors shown in the web version of this paper)

To tackle these visual clutters for better knowledge discovery, We innovate a quadratic energy reduction model to smoothly shape clusters against visual clutters while preserving essential details of each cluster, by associating each line i (with z_i being its center) between two adjacent dimension x and y with a "rubber band" effect with three potential energy:

Elastic Energy: $E_E(i) = (z_i - \frac{x_i+y_i}{2})^2$
Attraction Energy: $E_A(i, \hat{c}_p) = (z_i - \hat{c}_p)^2$
Repelling Energy: $E_R(i, \hat{c}_{p-1}, \hat{c}_{p+1}) = (z_i - \hat{c}_{p-1})^2 + (z_i - \hat{c}_{p+1})^2$

Here each cluster has an attracting center $\hat{c}_p$ which may serve as a repelling center for its adjacent clusters. We developed an efficient energy reduction model by properly initializing and manipulating $\hat{c}_p$ (omitted due to space limit).

The visualization effects are significantly enhanced by our energy reduction model. Figure 3(a) and Figure 3(b) are examples of enhanced visualization results for Figure 2(c) and (i), respectively, by our energy reduction models. Readers can easily observe more clusters and thus better understanding their relationships. It is easy to see the essential details of these clusters are not altered. More specifically, if two clusters are negatively-related or positively-related, the relationship not only remains after energy reduction, but gets further enhanced for human observation. For example, we can observe the blue cluster is negatively related to the pink cluster between the last two columns in Figure 3(a) while it is almost impossible to see this in Figure 2(c). As another example, the negative relation between cyan cluster and the light green cluster is more manifest in Figure 3(b) than in Figure 2(i). Finally, instead of affecting the observation of cluster interactions, outliers of each cluster can be easily identified as those few lines far away from the majority of lines.

In summary, given an order of coordinates, our energy reduction model efficiently provides better views of clusters for visual knowledge discovery at both the macro level (i.e., cluster interactions) and the micro level (i.e., individual lines with outliers clearly exposed).

5 Conclusion and Future Work

In this paper, we show a novel method to visualize discovered clusters in parallel coordinates. First, we provide good orders of coordinates for different knowledge discovery purposes. Second, we shape the clusters with a quadratic energy reduction model, such that cluster interactions are much easier to observe without compromising their essential details. Our empirical study on visualizing real datasets confirms that our method is effective and efficient. Our visualization techniques can further be combined with other visualization tools for better results, e.g, applying various visual rendering algorithms to enhance our visualization effects.

Acknowledgement. This work was supported by the US National Science Foundation under Grant #1019343 to the Computing Research Association for the CIFellows Project.

References

1. Ankerst, M., Berchtold, S., Keim, D.A.: Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In: IEEE Symposium on Information Visualization (INFOVIS), p. 52 (1998)
2. Artero, A.O., de Oliveira, M.C.F., Levkowitz, H.: Uncovering clusters in crowded parallel coordinates visualizations. In: IEEE Symposium on Information Visualization (INFOVIS), pp. 81–88 (2004)
3. Berthold, M.R., Hall, L.O.: Visualizing fuzzy points in parallel coordinates. IEEE Transactions on Fuzzy Systems 11(3), 369–374 (2003)
4. Böckenhauer, H.-J., Hromkovič, J., Kneis, J., Kupke, J.: On the Approximation Hardness of Some Generalizations of TSP. In: Arge, L., Freivalds, R. (eds.) SWAT 2006. LNCS, vol. 4059, pp. 184–195. Springer, Heidelberg (2006)
5. Christofides, N.: Worst-case analysis of a new heuristic for the travelling salesman problem. Graduate School of Industrial Administration, CMU, Report 388 (1976)
6. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. The MIT Press (2001)
7. Dasgupta, A., Kosara, R.: Pargnostics: Screen-Space Metrics for Parallel Coordinates. IEEE Transactions on Visualization and Computer Graphics 16(6), 1017–1026 (2010)
8. Ellis, G., Dix, A.: Enabling automatic clutter reduction in parallel coordinate plots. IEEE Transactions on Visualization and Computer Graphics 12, 717–724 (2006)
9. Fua, Y.-H., Ward, M.O., Rundensteiner, E.A.: Hierarchical parallel coordinates for exploration of large datasets. IEEE Visualization, 43–50 (1999)
10. Gross, J.L., Yellen, J.: Graph theory and its applications. CRC Press (2006)
11. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann (2000)
12. Hassin, R., Rubinstein, S.: A 7/8-approximation algorithm for metric max tsp. Inf. Process. Lett. 81(5), 247–251 (2002)
13. Holten, D., Van Wijk, J.J.: Evaluation of Cluster Identification Performance for Different PCP Variants. Computer Graphics Forum 29(3), 793–802 (2010)
14. Hoogeveen, J.A.: Analysis of christofides' heuristic: Some paths are more difficult than cycles. Operations Research Letters 10(5), 291–295 (1991)
15. Hurley, C.B.: Clustering visualizations of multidimensional data. Journal of Computational and Graphical Statistics 13(4), 788–806 (2004)
16. Hurley, C.B., Oldford, R.W.: Pairwise display of high-dimensional information via eulerian tours and hamiltonian decompositions. Journal of Computational and Graphical Statistics 19(4), 861–886 (2010)

17. Hurley, C.B., Oldford, R.W.: Eulerian tour algorithms for data visualization and the pairviz package. Computational Statistics 26(4), 613–633 (2011)
18. Inselberg, A.: The plane with parallel coordinates. The Visual Computer 1(2), 69–91 (1985)
19. Jain, A.K., Narasimha Murty, M., Flynn, P.J.: Data clustering: A review. ACM Comput. Surv. 31(3), 264–323 (1999)
20. Johansson, J., Ljung, P., Jern, M., Cooper, M.: Revealing structure within clustered parallel coordinates displays. In: IEEE Symposium on Information Visualization (INFOVIS), p. 17 (2005)
21. Kendall, M.G.: A new measure of rank correlation. Biometrika 30(1/2), 81–93 (1938)
22. Knight, W.R.: A computer method for calculating kendall's tau with ungrouped data. Journal of the American Statistical Association, 436–439 (1966)
23. Kosara, R., Bendix, F., Hauser, H.: Parallel sets: Interactive exploration and visual analysis of categorical data. IEEE Trans. Vis. Comput. Graph. 12(4), 558–568 (2006)
24. Laporte, G.: The traveling salesman problem: An overview of exact and approximate algorithms. European Journal of Operational Research 59(2), 231–247 (1992)
25. Moustafa, R., Wegman, E.: Multivariate Continuous Data - Parallel Coordinates. Springer, New York (2006)
26. Nelson, R.B.: Kendall tau metric. Encyclopaedia of Mathematics 3, 226–227 (2001)
27. Peng, W., Ward, M.O., Rundensteiner, E.A.: Clutter reduction in multidimensional data visualization using dimension reordering. In: IEEE Symposium on Information Visualization (INFOVIS), pp. 89–96 (2004)
28. Siirtola, H., Räihä, K.J.: Interacting with parallel coordinates. Interacting with Computers 18(6), 1278–1309 (2006)
29. Wegman, E.J.: The grand tour in k-dimensions. In: Computing Science and Statistics: Proceedings of the 22nd Symposium on the Interface, pp. 127–136 (1991)
30. Wegman, E.J.: Hyperdimensional data analysis using parallel coordinates. Journal of the American Statistical Association 85(411), 664–675 (1990)
31. Wegman, E.J.: Visual data mining. Statistics in Medicine 22, 1383–1397 (2003)
32. Wilhelm, A.F.X., Wegman, E.J., Symanzik, J.: Visual clustering and classification: The oronsay particle size data set revisited. Computational Statistics 14, 109–146 (1999)
33. Wong, P.C., Bergeron, R.D.: 30 years of multidimensional multivariate visualization. Scientific Visualization, 3–33 (1994)
34. Yang, J., Peng, W., Ward, M.O., Rundensteiner, E.A.: Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In: INFOVIS (2003)
35. Zhao, K., Liu, B., Tirpak, T.M., Schaller, A.: Detecting patterns of change using enhanced parallel coordinates visualization. In: ICDM, p. 747 (2003)
36. Zhou, H., Yuan, X., Qu, H., Cui, W., Chen, B.: Visual clustering in parallel coordinates. Comput. Graph. Forum 27(3), 1047–1054 (2008)

Feature Enriched Nonparametric Bayesian Co-clustering

Pu Wang, Carlotta Domeniconi, Huzefa Rangwala, and Kathryn B. Laskey

George Mason University
4400 University Ave., Fairfax, VA 22030 USA
{pwang7,carlotta,rangwala}@cs.gmu.edu, klaskey@gmu.edu

Abstract. Co-clustering has emerged as an important technique for mining relational data, especially when data are sparse and high-dimensional. Co-clustering simultaneously groups the different kinds of objects involved in a relation. Most co-clustering techniques typically only leverage the entries of the given contingency matrix to perform the two-way clustering. As a consequence, they cannot predict the interaction values for new objects. In many applications, though, additional features associated to the objects of interest are available. The *Infinite Hidden Relational Model* (IHRM) has been proposed to make use of these features. As such, IHRM has the capability to forecast relationships among previously unseen data. The work on IHRM lacks an evaluation of the improvement that can be achieved when leveraging features to make predictions for unseen objects. In this work, we fill this gap and re-interpret IHRM from a co-clustering point of view. We focus on the empirical evaluation of forecasting relationships between previously unseen objects by leveraging object features. The empirical evaluation demonstrates the effectiveness of the feature-enriched approach and identifies the conditions under which the use of features is most useful, i.e., with sparse data.

Keywords: Bayesian Nonparametrics, Dirichlet Processes, Co-clustering, Protein-molecule interaction data.

1 Introduction

Co-clustering [11] has emerged as an important approach for mining relational data. Often, data can be organized in a matrix, where rows and columns present a symmetrical relation. Co-clustering simultaneously groups the different kinds of objects involved in a relation; for example, proteins and molecules indexing a contingency matrix that holds information about their interaction. Molecules are grouped based on their binding patterns to proteins; similarly, proteins are clustered based on the molecules they interact with. The two clustering processes are inter-dependent. Understanding these interactions provides insight into the underlying biological processes and is useful for designing therapeutic drugs.

Existing co-clustering techniques typically only leverage the entries of the given contingency matrix to perform the two-way clustering. As a consequence,

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 517–529, 2012.

they cannot predict the interaction values for new objects. This greatly limits the applicability of current co-clustering approaches.

In many applications additional features associated to the objects of interest are available, e.g., sequence information for proteins. Such features can be leveraged to perform predictions on new data. The *Infinite Hidden Relational Model* (IHRM) [36] has been proposed to leverage features associated to the rows and columns of the contingency matrix to forecast relationships among previously unseen data. Although IHRM was originally introduced from a relational learning point of view, it is essentially a co-clustering model that overcomes the aforementioned limitations of existing co-clustering techniques. In particular, IHRM is a nonparametric Bayesian model, which learns the number of row and column clusters from the given samples. This is achieved by assuming Dirichlet Process priors to the rows and columns of the contingency matrix. As such, IHRM does not require the *a priori* specification of the numbers of row and column clusters in the data.

Existing Bayesian co-clustering models [30,35,19] are related to IHRM, but none makes use of features associated to the rows and columns of the contingency matrix. As a consequence, these methods can handle missing entries only for already observed rows and columns (e.g., for a protein and a molecule used during training, although not necessarily in combination). In particular, IHRM can be viewed as an extension to the nonparametric Bayesian co-clustering (NBCC) model [19]. IHRM adds to NBCC the ability to exploit features associated to rows and columns, thus enabling IHRM to predict entries for unseen rows and/or columns. The authors in [36] have applied IHRM to collaborative filtering [27]. Co-clustering techniques have also been applied to collaborative filtering [33,15,10], but again none of these involve features associated to rows or columns of the data matrix.

The work on IHRM [36] lacks an evaluation of the improvement that can be achieved when leveraging features to make predictions for unseen objects. In this work, we fill this gap and re-interpret IHRM from a co-clustering point of view. We call the resulting method Feature Enriched Dirichlet Process Co-clustering (FE-DPCC). We focus on the empirical evaluation of forecasting relationships between previously unseen objects by leveraging object features.

2 Related Work

Researchers have proposed several discriminative and generative co-clustering models, e.g. [7,29]. Bayesian Co-clustering (BCC) [30] maintains separate Dirichlet priors for row- and column-cluster probabilities. To generate an entry in the data matrix, the model first generates the row and column clusters for the entry from their respective Dirichlet-multinomial distributions. The entry is then generated from a distribution specific to the row- and column-cluster. Like the original Latent Dirichlet Allocation (LDA) [5] model, BCC assumes symmetric Dirichlet priors for the data distributions given the row- and column-clusters. Shan and Banerjee [30] proposed a variational Bayesian algorithm to perform inference with the BCC model. In [35], the authors proposed a variation of BCC, and developed a collapsed Gibbs sampling and a collapsed variational algorithm to perform inference. All aforementioned co-clustering models are parametric, i.e., they need to have specified the number of row- and column-clusters.

A nonparametric Bayesian co-clustering (NBCC) approach has been proposed in [19]. NBCC assumes two independent Bayesian priors on rows and columns. As such, NBCC does not require *a priori* the number of row- and column-clusters. NBCC assumes a Pitman-Yor Process [24] prior, which generalizes the Dirichlet Process. The feature-enriched method we introduce here is an extension of NBCC, where features associated to rows and columns are used. Such features enable our technique to predict entries for unseen rows/columns.

A related work is Bayesian matrix factorization. In [17], the authors alleviated overfitting in singular value decomposition (SVD) by specifying a prior distribution over parameters, and performing variational inference. In [26], the authors proposed a Bayesian probabilistic matrix factorization method, that assigns a prior distribution to the Gaussian parameters involved in the model. These Bayesian approaches to matrix factorization are parametric. Nonparametric Bayesian matrix factorization models include [8,32,25].

Our work is also related to collaborative filtering (CF) [27]. CF learns the relationships between users and items using only user preferences to items, and then recommends items to users based on the learned relationships. Various approaches have been proposed to discover underlying patterns in user consumption behaviors [6,16,1,18,17,26,31,12,14]. Co-clustering techniques have already been applied to CF [33,15,10]. None of these techniques involve features associated to rows or columns of the data matrix. On the contrary, content-based (CB) recommendation systems [3] predict user preferences to items using user and item features. In practice, CB methods are usually combined with CF. The approach we introduce in this paper is a Bayesian combination of CF and CB.

3 Background: Dirichlet Process

The Dirichlet process (DP) [9] is an infinite-dimensional generalization of the Dirichlet distribution. Formally, let S be a set, G_0 a measure on S, and α_0 a positive real number. The random probability distribution G on S is distributed as a DP with concentration parameter α_0 (also called the pseudo-count) and base measure G_0 if, for any finite partition $\{B_k\}_{1\leq k\leq K}$ of S: $(G(B_1), G(B_2), \cdots, G(B_K)) \sim \mathrm{Dir}(\alpha_0 G_0(B_1), \alpha_0 G_0(B_2), \cdots, \alpha_0 G_0(B_K))$.

Let G be a sample drawn from a DP. Then with probability 1, G is a discrete distribution [9]. Further, if the first $N-1$ draws from G yield K distinct values $\theta^*_{1:K}$ with multiplicities $n_{1:K}$, then the probability of the N^{th} draw conditioned on the previous $N-1$ draws is given by the Pólya urn scheme [4]:

$$\theta_N = \begin{cases} \theta^*_k, & \text{with prob } \frac{n_k}{N-1+\alpha_0},\ k \in \{1, \cdots, K\} \\ \theta^*_{K+1} \sim G_0, & \text{with prob } \frac{\alpha_0}{N-1+\alpha_0} \end{cases}$$

The DP is often used as a nonparametric prior in Bayesian mixture models [2]. Assume the data are generated from the following generative procedure: $G \sim \text{Dir}(\alpha_0, G_0); \theta_{1:N} \sim G; x_{1:N} \sim \prod_{n=1}^{N} F(\cdot|\theta_n)$, where the $F(\cdot|\theta_n)$ are probability distributions known as mixture components. Typically, there are duplicates among the $\theta_{1:N}$; thus, multiple data points are generated from the same mixture component. It is natural to define a cluster as those observations generated from a given mixture component. This model is known as the *Dirichlet process mixture* (DPM) model. Although any finite sample contains only finitely many clusters, there is no bound on the number of clusters and any new data point has non-zero probability of being drawn from a new cluster [20]. Therefore, DPM is known as an "infinite" mixture model.

The DP can be generated via the stick-breaking construction [28]. Stick-breaking draws two infinite sequences of independent random variables, $v_k \sim \text{Beta}(1, \alpha_0)$ and $\theta_k^* \sim G_0$ for $k = \{1, 2, \cdots\}$. Let G be defined as:

$$\pi_k = v_k \prod_{j=1}^{k-1} (1 - v_j) \qquad\qquad G = \sum_{k=1}^{\infty} \pi_k \delta(\theta_k^*) \tag{1}$$

where $\boldsymbol{\pi} = \langle \pi_k | k = 1, 2, \cdots \rangle$ are mixing proportions and $\delta(\theta)$ is the distribution that samples the value θ with probability 1. Then $G \sim \text{Dir}(\alpha_0, G_0)$. It is helpful to use an indicator variable z_n to denote which mixture component is associated with x_n. The generative process for the DPM model using stick-breaking is as follows (additional details on the DPM model can be found in [20,23]):

1. Draw $v_k \sim \text{Beta}(1, \alpha_0)$, $k = \{1, 2, \cdots\}$ and calculate $\boldsymbol{\pi}$ as in Eq (1).
2. Draw $\theta_k^* \sim G_0$, $k = \{1, 2, \cdots\}$
3. For each data point $n = \{1, 2, \cdots, N\}$:
   - Draw $z_n \sim \text{Discrete}(\boldsymbol{\pi})$; Draw $x_n \sim F(\cdot|\theta_{z_n}^*)$

4 Feature Enriched Dirichlet Process Co-clustering

The observed data $\boldsymbol{X}$ of FE-DPCC are composed of three parts: the observed row features $\boldsymbol{X}^R$, the observed column features $\boldsymbol{X}^C$, and the observed relational features $\boldsymbol{X}^E$ between rows and columns. If there are R rows and C columns, then $\boldsymbol{X}^R = \langle \boldsymbol{x}_r^R | r = \{1, \cdots, R\} \rangle$, $\boldsymbol{X}^C = \langle \boldsymbol{x}_c^C | c = \{1, \cdots, C\} \rangle$, and $\boldsymbol{X}^E = \langle x_{rc}^E | r = \{1, \cdots, R\}, c = \{1, \cdots, C\} \rangle$. $\boldsymbol{X}^E$ may have missing data, i.e., some entries may not be observed.

FE-DPCC is a generative model that assumes two independent DPM priors on rows and columns. We follow a stick-breaking representation to describe the FE-DPCC model. Specifically, assuming row and column DP priors $\text{Dir}(\alpha_0^R, G_0^R)$ and $\text{Dir}(\alpha_0^C, G_0^C)$, FE-DPCC draws row-cluster parameters $\boldsymbol{\theta}_k^{*R}$ from G_0^R, for $k = \{1, \cdots, \infty\}$, column-cluster parameters $\boldsymbol{\theta}_l^{*C}$ from G_0^C, for $l = \{1, \cdots, \infty\}$, and co-cluster parameters $\boldsymbol{\theta}_{kl}^{*E}$ from

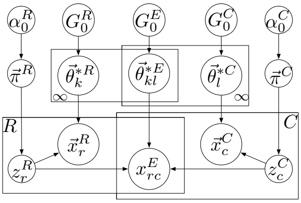


Fig. 1. FE-DPCC model

G_0^E, for each combination of k and l[1]; then draws row mixture proportion $\boldsymbol{\pi}^R$ and column mixture proportion $\boldsymbol{\pi}^C$ as defined in Eq. 1. For each row r and each column c, FE-DPCC draws the row-cluster indicator z_r^R and column-cluster indicator z_c^C according to $\boldsymbol{\pi}^R$ and $\boldsymbol{\pi}^C$, respectively. Further, FE-DPCC assumes the observed features of each row r and each column c are drawn from two parametric distributions $F(\cdot|\boldsymbol{\theta}_k^{*R})$ and $F(\cdot|\boldsymbol{\theta}_l^{*C})$, respectively, and each entry, x_{rc}^E, of the relational feature matrix is drawn from a parametric distribution $F(\cdot|\boldsymbol{\theta}_{kl}^{*E})$, where $z_r^R = k$ and $z_c^C = l$.

The generative process for FE-DPCC is as follows and the FE-DPCC model is illustrated in Figure 1.

1. Draw $v_k^R \sim \text{Beta}(1, \alpha_0^R)$, for $k = \{1, \cdots, \infty\}$ and calculate $\boldsymbol{\pi}^R$ as in Eq (1)
2. Draw $\boldsymbol{\theta}_k^{*R} \sim G_0^R$, for $k = \{1, \cdots, \infty\}$
3. Draw $v_l^C \sim \text{Beta}(1, \alpha_0^C)$, for $l = \{1, \cdots, \infty\}$ and calculate $\boldsymbol{\pi}^C$ as in Eq (1)
4. Draw $\boldsymbol{\theta}_l^{*C} \sim G_0^C$, for $l = \{1, \cdots, \infty\}$
5. Draw $\boldsymbol{\theta}_{kl}^{*E} \sim G_0^E$, for $k = \{1, \cdots, \infty\}$ and $l = \{1, \cdots, \infty\}$
6. For each row $r = \{1, \cdots, R\}$, draw $z_r^R \sim \text{Discrete}(\boldsymbol{\pi}^R)$, and draw $\boldsymbol{x}_r^R \sim F(\cdot|\boldsymbol{\theta}_{z_r^R}^{*R})$
7. For each column $c = \{1, \cdots, C\}$, draw $z_c^C \sim \text{Discrete}(\boldsymbol{\pi}^C)$, and draw $\boldsymbol{x}_c^C \sim F(\cdot|\boldsymbol{\theta}_{z_c^C}^{*C})$
8. For each entry $\boldsymbol{x}_{rc}^E$, draw $\boldsymbol{x}_{rc}^E \sim F(\cdot|\boldsymbol{\theta}_{z_r^R z_c^C}^{*E})$

4.1 Inference

The likelihood of the observed data is given by:

$$p(\boldsymbol{X}|\boldsymbol{Z}^R, \boldsymbol{Z}^C, \boldsymbol{\theta}^{*R}, \boldsymbol{\theta}^{*C}, \boldsymbol{\theta}^{*E}) = (\prod_{r=1}^{R} f(\boldsymbol{x}_r^R|\boldsymbol{\theta}_{z_r^R}^{*R}))(\prod_{c=1}^{C} f(\boldsymbol{x}_c^C|\boldsymbol{\theta}_{z_c^C}^{*C}))(\prod_{r=1}^{R}\prod_{c=1}^{C} f(x_{rc}^E|\boldsymbol{\theta}_{z_r^R z_c^C}^{*E}))$$

where $f(\cdot|\boldsymbol{\theta}_k^{*R})$, $f(\cdot|\boldsymbol{\theta}_l^{*C})$ and $f(\cdot|\boldsymbol{\theta}_{kl}^{*E})$ denote the probability density (or mass) functions of $F(\cdot|\boldsymbol{\theta}_k^{*R})$, $F(\cdot|\boldsymbol{\theta}_l^{*C})$ and $F(\cdot|\boldsymbol{\theta}_{kl}^{*E})$, respectively; $\boldsymbol{Z}^R = \langle z_r^R | r = \{1, \cdots, R\}\rangle$; $\boldsymbol{Z}^C = \langle z_c^C | c = \{1, \cdots, C\}\rangle$; $\boldsymbol{\theta}^{*R} = \langle \boldsymbol{\theta}_k^{*R} | k = \{1, \cdots, \infty\}\rangle$; $\boldsymbol{\theta}^{*C} = \langle \boldsymbol{\theta}_l^{*C} | l = \{1, \cdots, \infty\}\rangle$; and $\boldsymbol{\theta}^{*E} = \langle \boldsymbol{\theta}_{kl}^{*E} | k = \{1, \cdots, \infty\}, l = \{1, \cdots, \infty\}\rangle$.

The marginal likelihood obtained by integrating out the model parameters $\boldsymbol{\theta}^{*R}$, $\boldsymbol{\theta}^{*C}$, and $\boldsymbol{\theta}^{*E}$ is:

$$p(\boldsymbol{X}|\boldsymbol{Z}^R, \boldsymbol{Z}^C, G_0^R, G_0^C, G_0^E) = \left(\prod_{r=1}^{R} \int f(\boldsymbol{x}_r^R|\boldsymbol{\theta}_{z_r^R}^{*R}) g(\boldsymbol{\theta}_{z_r^R}^{*R}|\zeta^R) d\boldsymbol{\theta}_{z_r^R}^{*R}\right) \tag{2}$$

$$\left(\prod_{c=1}^{C} \int f(\boldsymbol{x}_c^C|\boldsymbol{\theta}_{z_c^C}^{*C}) g(\boldsymbol{\theta}_{z_c^C}^{*C}|\zeta^C) d\boldsymbol{\theta}_{z_c^C}^{*C}\right) \left(\prod_{r=1}^{R}\prod_{c=1}^{C} \int f(x_{rc}^E|\boldsymbol{\theta}_{z_r^R z_c^C}^{*E}) g(\boldsymbol{\theta}_{z_r^R z_c^C}^{*E}|\zeta^E) d\boldsymbol{\theta}_{z_r^R z_c^C}^{*E}\right)$$

where $g(\cdot|\zeta^R)$, $g(\cdot|\zeta^C)$ and $g(\cdot|\zeta^E)$ denote the probability density functions of G_0^R, G_0^C and G_0^E, respectively. We assume $F(\cdot|\boldsymbol{\theta}_k^{*R})$ and G_0^R, $F(\cdot|\boldsymbol{\theta}_l^{*C})$ and G_0^C, and $F(\cdot|\boldsymbol{\theta}_{kl}^{*E})$ and G_0^E are all pairwise conjugate. Thus, there is a closed form expression for the marginal likelihood (2). The conditional distribution for sampling the row-cluster indicator variable z_r^R for the r^{th} row $\boldsymbol{x}_r^R$ is as follows. For populated row-clusters $k \in \{Z_{r'}^R\}_{r'=\{1,\cdots,r-1,r+1,\cdots,R\}}$,

[1] Every co-cluster is indexed by a row-cluster ID and a column-cluster ID. Thus, we denote a co-cluster defined by the k^{th} row-cluster and the l^{th} column-cluster as (k, l).

$$p(z_r^R = k|\boldsymbol{x}_r^R, \{x_{rc}^E\}_{c\in\{1,\cdots,C\}}, \boldsymbol{X}^{R\neg r}, \boldsymbol{X}^{E\neg r}, \boldsymbol{Z}^{R\neg r}) \propto \tag{3}$$
$$\frac{\mathcal{N}_k^{\neg r}}{R-1+\alpha_0^R}\left(\int f(\boldsymbol{x}_r^R|\boldsymbol{\theta}_k^{*R})g(\boldsymbol{\theta}_k^{*R}|\zeta_k^{*R\neg r})d\boldsymbol{\theta}_k^{*R}\right)\prod_{c=1}^{C}\left(\int f(x_{rc}^E|\boldsymbol{\theta}_{kz_c^C}^{*E})g(\boldsymbol{\theta}_{kz_c^C}^{*E}|\zeta_{kz_c^C}^{*E\neg r})d\boldsymbol{\theta}_{kz_c^C}^{*E}\right)$$

where $\neg r$ means excluding the r^{th} row, $\mathcal{N}_k^{\neg r}$ is the number of rows assigned to the k^{th} row-cluster excluding the r^{th} row, $\zeta_k^{*R\neg r}$ is the hyperparameter of the posterior distribution of the k^{th} row-cluster parameter $\boldsymbol{\theta}_k^{*R}$ given all rows assigned to the k^{th} row-cluster excluding the r^{th} row, and $\zeta_{kz_c^C}^{*E\neg r}$ is the hyperparameter of the posterior distribution of the co-cluster (k, z_c^C) given all entries assigned to it excluding the entries in the r^{th} row. When $k \notin \{z_{r'}^R\}_{r'=\{1,\cdots,r-1,r+1,\cdots,R\}}$, i.e., z_r^R is being set to its own singleton row-cluster, the conditional distribution becomes:

$$p(z_r^R = k|\boldsymbol{x}_r^R, \{x_{rc}^E\}_{c\in\{1,\cdots,C\}}, \boldsymbol{X}^{R\neg r}, \boldsymbol{X}^{E\neg r}, \boldsymbol{Z}^{R\neg r}) \propto \tag{4}$$
$$\frac{\alpha_0^R}{R-1+\alpha_0^R}\left(\int f(\boldsymbol{x}_r^R|\boldsymbol{\theta}_k^{*R})g(\boldsymbol{\theta}_k^{*R}|\zeta^R)d\boldsymbol{\theta}_k^{*R}\right)\prod_{c=1}^{C}\left(\int f(x_{rc}^E|\boldsymbol{\theta}_{kz_c^C}^{*E})g(\boldsymbol{\theta}_{kz_c^C}^{*E}|\zeta_{kz_c^C}^{*E\neg r})d\boldsymbol{\theta}_{kz_c^C}^{*E}\right)$$

The conditional distribution for sampling the column-cluster indicator variable z_c^C for the c^{th} column $\boldsymbol{x}_c^C$ is obtained analogously. For populated column-clusters $l \in \{Z_{c'}^C\}_{c'=\{1,\cdots,c-1,c+1,\cdots,C\}}$,

$$p(z_c^C = l|\boldsymbol{x}_c^C, \{x_{rc}^E\}_{r\in\{1,\cdots,R\}}, \boldsymbol{X}^{C\neg c}, \boldsymbol{X}^{E\neg c}, \boldsymbol{Z}^{C\neg c}) \propto \tag{5}$$
$$\frac{\mathcal{N}_l^{\neg c}}{C-1+\alpha_0^C}\left(\int f(\boldsymbol{x}_c^C|\boldsymbol{\theta}_l^{*C})g(\boldsymbol{\theta}_l^{*C}|\zeta_l^{*C\neg c})d\boldsymbol{\theta}_l^{*C}\right)\prod_{r=1}^{R}\left(\int f(x_{rc}^E|\boldsymbol{\theta}_{z_r^R l}^{*E})g(\boldsymbol{\theta}_{z_r^R l}^{*E}|\zeta_{z_r^R l}^{*E\neg c})d\boldsymbol{\theta}_{z_r^R l}^{*E}\right)$$

where $\neg c$ means excluding the c^{th} column, $\mathcal{N}_l^{\neg c}$ is the number of columns assigned to the l^{th} column-cluster excluding the c^{th} column, $\zeta_l^{*C\neg c}$ is the hyperparameter of the posterior distribution of the l^{th} column-cluster parameter $\boldsymbol{\theta}_l^{*C}$ given all columns assigned to the l^{th} column-cluster excluding the c^{th} column, and $\zeta_{z_r^R l}^{*E\neg c}$ is the hyperparameter of the posterior distribution of the co-cluster (z_r^R, l) given all entries assigned to it excluding the entries in the c^{th} column. If $z_c^C \notin \{z_{c'}^C\}_{c'=\{1,\cdots,c-1,c+1,\cdots,C\}}$, i.e., z_c^C is being assigned to its own singleton column-cluster, the conditional distribution becomes:

$$p(z_c^C = l|\boldsymbol{x}_c^C, \{x_{rc}^E\}_{r\in\{1,\cdots,R\}}, \boldsymbol{X}^{C\neg c}, \boldsymbol{X}^{E\neg c}, \boldsymbol{Z}^{C\neg c}) \propto \tag{6}$$
$$\frac{\alpha_0^C}{C-1+\alpha_0^C}\left(\int f(\boldsymbol{x}_c^C|\boldsymbol{\theta}_l^{*C})g(\boldsymbol{\theta}_l^{*C}|\zeta^C)d\boldsymbol{\theta}_l^{*C}\right)\prod_{r=1}^{R}\left(\int f(x_{rc}^E|\boldsymbol{\theta}_{z_r^R l}^{*E})g(\boldsymbol{\theta}_{z_r^R l}^{*E}|\zeta_{z_r^R l}^{*E\neg c})d\boldsymbol{\theta}_{z_r^R l}^{*E}\right)$$

5 Experimental Evaluation

We conducted experiments on two rating datasets and two protein-molecule interaction datasets. MovieLens[2] is a movie recommendation dataset containing 100,000 ratings in a sparse data matrix for 1682 movies rated by 943 users. Jester[3] is a joke rating dataset. The original dataset contains 4.1 million continuous ratings of 140 jokes from 73,421 users. We chose a subset containing 100,000 ratings. Following [30], we uniformly discretized the ratings into 10 bins.

[2] http://www.grouplens.org/node/73

[3] http://goldberg.berkeley.edu/jester-data/

We also used two protein-molecule interaction datasets. The first dataset (MP1[4]) consists of G-protein coupled receptor (GPCR) proteins and their interaction with small molecules [13]. These interactions are the product of an experiment that determines whether a particular protein target is modulated by a molecule. MP1 had 4051 interactions between 166 proteins and 2687 molecules. The second dataset (MP2[5]) [21] differs from MP1 in that the protein targets belong to a more general class and are not restricted to GPCRs. The use of targets restricted to a specific group of proteins (GPCRs) is similar to a *chemogenomics* approach where the assumption is that proteins in the same family have a similar activity/interaction profile. MP2 had 154 proteins, 2876 molecules and a total of 7146 positive interactions. Table 1 summarizes the dataset characteristics.

5.1 Experimental Methodology and Feature Information

Table 1. Training and Test Data

		MovieLens	Jester	MP1	MP2
Train	# Rows	943	33459	1961	2674
	# Columns	1650	140	61	149
	# Entries	80000	80000	3000	5000
	Density	5.142%	1.708%	2.508%	1.255%
Test	# Rows	927	14523	856	1647
	# Columns	1407	139	68	145
	# Entries	20000	20000	1051	2146
	Density	1.533%	0.991%	1.806%	0.899%

We first compared FE-DPCC with a variant of NBCC, called *Dirichlet Process Co-clustering* (DPCC). DPCC restricts the Pitman-Yor priors of NBCC to the special case of independent Dirichlet Process priors on rows and columns, so as to compare with FE-DPCC fairly. So, the difference between FE-DPCC and DPCC is that FE-DPCC augments DPCC to exploit row and column features. We ran 1000 iterations of Gibbs sampling for both FE-DPCC and DPCC. We used perplexity as an evaluation metric to compare FE-DPCC with DPCC on all the test data. The perplexity of a dataset D is defined as $perplexity(D) = \exp\left(-\mathcal{L}(D)/N\right)$, where $\mathcal{L}(D)$ is the log-likelihood of D, and N is the number of data points in D. The higher the log-likelihood, the lower the perplexity, and the better a model fits the data.

The relational features in our data are discrete. We assume $f(\cdot|\boldsymbol{\theta}^{*E}_{kl})$ is a categorical distribution, $\mathrm{Cat}(\cdot|\boldsymbol{\theta}^{*E}_{kl})$, and $g(\boldsymbol{\theta}^{*E}_{kl}|\zeta^E)$ is a Dirichlet distribution, $\mathrm{Dir}(\boldsymbol{\theta}^{*E}_{kl}|\boldsymbol{\varphi})$, with $\zeta^E = \boldsymbol{\varphi}$. Because of conjugacy, we can marginalize out $\boldsymbol{\theta}^{*E}_{kl}$. Without loss of generality, we assume that $f(\cdot|\boldsymbol{\theta}^{*E}_{kl})$ is a D-dimensional categorical distribution with support $\{1, \cdots, D\}$, and we denote the Dirichlet hyperparameter as $\zeta^E = \boldsymbol{\varphi} = \langle \varphi_d | d = \{1, \cdots, D\}\rangle$. The predictive distribution of the co-cluster (k, l) to observe a new entry $x^E_{r'c'} = d$, $d \in \{1, \cdots, D\}$, is:

$$p(x^E_{r'c'} = d|\zeta^{*E}_{kl}, z^R_{r'} = k, z^C_{c'} = l) = \int f(x^E_{r'c'} = d|\boldsymbol{\theta}^{*E}_{kl}) g(\boldsymbol{\theta}^{*E}_{kl}|\zeta^{*E}_{kl}) d\boldsymbol{\theta}^{*E}_{kl} =$$

$$\int \mathrm{Cat}(x^E_{r'c'} = d|\boldsymbol{\theta}^{*E}_{kl}) \mathrm{Dir}(\boldsymbol{\theta}^{*E}_{kl}|\boldsymbol{\varphi}^*_{kl}) d\boldsymbol{\theta}^{*E}_{kl} \propto \mathcal{N}^d_{(k,l)} + \varphi_d$$

[4] http://pharminfo.pharm.kyoto-u.ac.jp/services/glida/

[5] http://pubchem.ncbi.nlm.nih.gov/

Table 2. Average Test Perplexity

		MovieLens	Jester	MP1	MP2
DPCC	Row and Column Observed	3.327 (0.020)	17.111 (0.031)	1.430 (0.011)	1.484 (0.013)
	Row or Column Unseen	4.427 (0.047)	19.322 (0.025)	8.845 (0.011)	7.987 (0.011)
	Overall Perplexity	4.424 (0.087)	18.116 (0.035)	8.843 (0.013)	7.980 (0.021)
FE-DPCC	Row and Column Observed	3.344 (0.021)	17.125 (0.040)	1.435 (0.024)	1.489 (0.023)
	Row or Column Unseen	3.892 (0.026)	17.836 (0.053)	1.453 (0.026)	1.509 (0.024)
	Overall Perplexity	3.889 (0.031)	17.836 (0.062)	1.450 (0.046)	1.501 (0.045)

where $\boldsymbol{\varphi}^*_{kl}$ is the posterior hyperparameter of the Dirichlet distribution of the co-cluster (k, l), and $\mathcal{N}^d_{(k,l)}$ is the number of entries assigned to the co-cluster (k, l) and is equal to d.

In MovieLens, users (rows) are represented with age, gender, and occupation, whereas the movies (columns) are associated with a 19-dimensional genre-representing binary vector. We assumed independence among the row features and the column features conditional on row- and column-clusters. We modeled age as drawn from a Poisson distribution, $\text{Poi}(\cdot|\lambda)$, with a conjugate Gamma prior, $\text{Gamma}(\lambda|\varrho, \varsigma)$. We modeled gender as drawn from a Bernoulli distribution, $\text{Ber}(\cdot|\vartheta)$, with a conjugate Beta prior $\text{Beta}(\vartheta|\varkappa, \varpi)$. The occupation feature is categorical, modeled as $\text{Cat}(\cdot|\boldsymbol{\phi})$, with Dirichlet prior, $\text{Dir}(\boldsymbol{\phi}|\boldsymbol{\varphi})$. Thus, the row feature parameter is given by $\boldsymbol{\theta}^{*R}_k = \langle \lambda^*_k, \vartheta^*_k, \boldsymbol{\phi}^*_k \rangle$, and the row feature prior hyperparameter is $\zeta^R = \langle \varrho, \varsigma, \vartheta, \boldsymbol{\varphi} \rangle$. We denote the feature vector of a new user as $\boldsymbol{x}^R_{r'} = \langle a_{r'}, g_{r'}, o_{r'} \rangle$, where $a_{r'}$, $g_{r'}$, and $o_{r'}$ represent the age, gender and occupation, respectively. The predictive distribution of the k^{th} row-cluster observing a new user, $\boldsymbol{x}^R_{r'}$, is:

$$p(\boldsymbol{x}^R_{r'}|\varrho^*_k, \varsigma^*_k, \varkappa^*_k, \varpi^*_k, \boldsymbol{\varphi}^*_k, z^R_{r'} = k) = \left(\int \text{Poi}(a_{r'}|\lambda^*_k) \text{Gamma}(\lambda^*_k|\varrho^*_k, \varsigma^*_k) d\lambda^*_k \right)$$
$$\left(\int \text{Ber}(g_{r'}|\vartheta^*_k) \text{Beta}(\vartheta^*_k|\varkappa^*_k, \varpi^*_k) d\vartheta^*_k \right) \left(\int \text{Cat}(o_{r'}|\boldsymbol{\phi}^*_k) \text{Dir}(\boldsymbol{\phi}^*_k|\boldsymbol{\varphi}^*_k) d\boldsymbol{\phi}^*_k \right) \qquad (7)$$

where ϱ^*_k, ς^*_k, $\varkappa^*_k$, ϖ^*_k, and $\boldsymbol{\varphi}^*_k$ are the posterior hyperparameters (k indexes the row-clusters). Denote $\zeta^{*R}_k = \langle \varrho^*_k, \varsigma^*_k, \varkappa^*_k, \varpi^*_k, \boldsymbol{\varphi}^*_k \rangle$. We assume that features associated with movies are generated from a Multinomial distribution, $\text{Mul}(\cdot|\boldsymbol{\psi})$, with Dirichlet prior, $\text{Dir}(\boldsymbol{\psi}|\boldsymbol{\varphi})$. Accordingly, $\boldsymbol{\theta}^{*C}_l = \boldsymbol{\psi}^*_l$, and $\zeta^C = \boldsymbol{\varphi}$. The predictive distribution of the l^{th} column-cluster observing a new movie, $\boldsymbol{x}^C_{c'}$, is: $p(\boldsymbol{x}^C_{c'}|\boldsymbol{\varphi}^*_l, z^C_{c'} = l) = \int \text{Mul}(\boldsymbol{x}^C_{c'}|\boldsymbol{\psi}^*_l) \text{Dir}(\boldsymbol{\psi}^*_l|\boldsymbol{\varphi}^*_l) d\boldsymbol{\psi}^*_l$, where $\zeta^{*C}_l = \boldsymbol{\varphi}^*_l$ is the posterior hyperparameter of the Dirichlet distribution (l indexes the column-clusters).

In Jester, there are no features associated with the users (rows), thus row-clusters cannot predict an unseen user. We used a bag-of-word representation for joke features, and assumed each joke feature vector is generated from a Multinomial distribution, $\text{Mul}(\cdot|\boldsymbol{\psi})$, with a Dirichlet prior, $\text{Dir}(\boldsymbol{\psi}|\boldsymbol{\varphi})$. The predictive distribution of the l^{th} column-cluster observing a new joke, $\boldsymbol{x}^C_{c'}$, is: $p(\boldsymbol{x}^C_{c'}|\boldsymbol{\varphi}^*_l, z^C_{c'} = l) = \int \text{Mul}(\boldsymbol{x}^C_{c'}|\boldsymbol{\psi}^*_l) \text{Dir}(\boldsymbol{\psi}^*_l|\boldsymbol{\varphi}^*_l) d\boldsymbol{\psi}^*_l$.

For MP1 and MP2, rows represent molecules and columns represent proteins. We extracted k-mer features from protein sequences. For MP1, we also used hierarchical features for proteins obtained from annotation databases. We used a graph-fragment-based feature representation that computes the frequency of different length cycles and paths for each molecule. These graph-fragment-based features were derived using AFGEN [34] (default parameters were used), known to capture structural aspects of molecules effectively. We assumed each protein was generated from a Multinomial distribution, $\text{Mul}(\cdot|\boldsymbol{\psi}^p)$, with a Dirichlet prior, $\text{Dir}(\boldsymbol{\psi}^p|\boldsymbol{\varphi}^p)$. We also assumed each molecule was generated from a Multinomial distribution, $\text{Mul}(\cdot|\boldsymbol{\psi}^m)$, with a Dirichlet prior, $\text{Dir}(\boldsymbol{\psi}^m|\boldsymbol{\varphi}^m)$. The predictive distribution of the k^{th} row-cluster observing a new molecule, $\boldsymbol{x}^R_{r'}$, is: $p(\boldsymbol{x}^R_{r'}|\boldsymbol{\varphi}^{*m}_k, z^R_{r'} = k) = \int \text{Mul}(\boldsymbol{x}^R_{r'}|\boldsymbol{\psi}^{*m}_k)\text{Dir}(\boldsymbol{\psi}^{*m}_k|\boldsymbol{\varphi}^{*m}_k)d\boldsymbol{\psi}^{*m}_k$.

The predictive distribution of the l^{th} column-cluster observing a new protein, $\boldsymbol{x}^C_{c'}$, is: $p(\boldsymbol{x}^C_{c'}|\boldsymbol{\varphi}^{*p}_l, z^C_{c'} = l) = \int \text{Mul}(\boldsymbol{x}^C_{c'}|\boldsymbol{\psi}^{*p}_l)\text{Dir}(\boldsymbol{\psi}^{*p}_l|\boldsymbol{\varphi}^{*p}_l)d\boldsymbol{\psi}^{*p}_l$.

5.2 Results

We performed a series of experiments to evaluate the performance of FE-DPCC across the four datasets. All experiments were repeated five times, and we report the average (and standard deviation) perplexity across the five runs. The experiments were performed on an Intel four core, Linux server with 4GB memory. The average running time for FE-DPCC was 1, 3, 3.5 and 2.5 hours on the MovieLens, Jester, MP1 and MP2 datasets, respectively.

Feature Enrichment Evaluation. Table 2 shows the average perplexity (and standard deviations) across five runs on the test data. To analyze the effect of new rows and columns on the prediction capabilities of the algorithms, we split each test set into subsets based on whether the subset contains new rows or columns w.r.t. the corresponding training data. Table 2 shows that the overall perplexity of FE-DPCC is lower than that of DPCC on all data, with an improvement of 12%, 1.5%, 84% and 81% for MovieLens, Jester, MP1 and MP2, respectively.

FE-DPCC is significantly better than DPCC on the portion of the test data that contains unseen rows and/or columns. These test sets consist of entries for rows and columns that are not included in the training set. The DPCC algorithm does not use features; as such it can predict entries for the new rows and columns using prior probabilities only. In contrast, the FE-DPCC algorithm leverages features along with prior probabilities; this enables our approach to predict values for the independent test entries more accurately. This ability is a major strength of our FE-DPCC algorithm. For the portion of the test data whose rows and columns are observed in the training as well, the perplexity values of FE-DPCC and DPCC are comparable. The standard deviations indicate that the algorithms are stable, yielding consistent results across different runs.

To accurately assess the performance of FE-DPCC, we performed a set of experiments that involved a perturbation of the protein and molecule features on MP1. Results are in Table 3. For these experiments, we used k-mer sequence features. First, we took the protein sequences (i.e., columns) and shuffled the ordering of the amino acids. This alters the ordering of the protein sequence but maintains the same composition (i.e., the shuffled sequences have the same

Table 3. Evaluation of feature enrichment on MP1

	Perplexity
Shuffle P	3.034 (0.083)
Exchange M	2.945 (0.083)
Exchange P	2.932 (0.071)
Exchange M&P	2.991 (0.095)
Use Only M	7.235 (0.043)
Use Only P	7.789 (0.045)
Use M and P	1.450 (0.046)

Table 4. Evaluation of protein features on MP1

	Perplexity
2-mer	1.471 (0.057)
3-mer	1.437 (0.044)
4-mer	1.441 (0.049)
5-mer	1.450 (0.046)
HF	1.413 (0.010)

Table 5. RMSE on Test Data

	FE-DPCC	*Slope One*
Movie	0.838 (0.031)	0.924 (0.035)
Jester	0.896 (0.062)	0.961 (0.065)

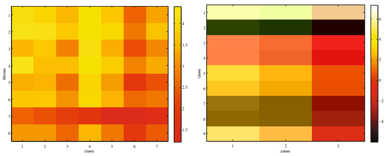
number of characters or amino acids). We refer to this scheme as "Shuffle". It achieves an average perplexity of 3.034, versus the average perplexity of 1.450 achieved by FE-DPCC (with no shuffling of features). We also devised a scheme in which the row and/or column features are exchanged, e.g., the features of a particular molecule are exchanged with the features of another molecule. Such an exchange causes the inclusion of incorrect information within the FE-DPCC algorithm. Our aim was to assess the strength of FE-DPCC when enriched with meaningful and correct features. We refer to this scheme as "Exchange." Table 3 shows the results of exchanging molecule features only (Exchange M), protein features only (Exchange P), and both (Exchange M and P). We noticed an average perplexity of 2.9 in each case. We also evaluated the FE-DPCC algorithm when only molecule or only protein features are used ("Use Only M" and "Use only P" in Table 3). The use of only one set of features prevents the co-clustering algorithm from making inferences on the unseen rows or columns in the test set.

For MP1 we performed additional experiments to evaluate the sequence features. The features are overlapping subsequences of a fixed length extracted from the protein sequences. We used k-mer lengths of 2, 3, 4 and 5, and observed that the average perplexity (Table 4) remained similar. As such, we used 5-mer features in all the experiments. We also compared the sequence features for the proteins to an alternate feature derived from a hierarchical biological annotation of the proteins. For MP1 the hierarchical features were extracted as done in the previous study [13,22]. From Table 4 we observe that the hierarchical features (HF) achieved a slightly lower perplexity as compared to the 5-mer features. This is encouraging, as it suggests that sequence features perform similarly to manual annotation (hierarchy), that may not be easily available for all the proteins.

Comparative Performance. We compared FE-DPCC with a well known collaborative filtering model, *Slope One* [16]. We used a Slope One implementation from the Apache Mahout machine learning library[6]. We used the root mean square error (RMSE) [6] to compare FE-DPCC and Slope One on MovieLens and Jester. Table 5 shows the RMSE values (and standard deviations) of FE-DPCC and Slope One across five runs on the test sets[7]. These results show

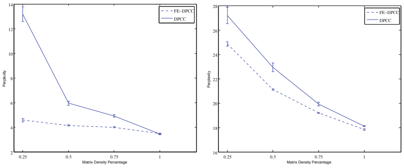
[6] http://mahout.apache.org/

[7] No new rows or columns in the test sets w.r.t. the training sets.



(a) MovieLens (b) Jester

Fig. 2. Co-clusters Learned by FE-DPCC



(a) MovieLens (b) Jester

Fig. 3. Test Perplexity with Different Densities

that incorporating row and column features is beneficial for the prediction of relationships.

Visualization of Co-clusters. In Figure 2 we illustrate the co-cluster structures learned by FE-DPCC on MovieLens and Jester. We calculate the mean entry value for each co-cluster, and plot the resulting mean values.

Data Density. We varied the density of MovieLens and Jester to see how it affects the perplexity of FE-DPCC and DPCC. We varied the matrix density by randomly sampling 25%, 50% and 75% of the entries in the training data. The sampled matrices were then given as input to DPCC and FE-DPCC to train a model and infer unknown entries on the test data. Figure 3 illustrates the results averaged across five iterations. As the sparsity of the relational matrix increases the test perplexity increases for both FE-DPCC and DPCC. But DPCC has far higher perplexity for a sparser matrix. As the matrix sparsity increases, the information within the relational matrix is lost and the FE-DPCC algorithm relies on the row and column features. Thus, for sparser matrices FE-DPCC shows far better results than DPCC. These experiments suggest the reason why we see a more dramatic difference between the two algorithms for MP1 and MP2, which are very sparse (see Table 1).

6 Conclusion

In this work, we focus on the empirical evaluation of FE-DPCC to predict relationships between previously unseen objects by using object features. We conducted experiments on a variety of relational data, including protein-molecule interaction data. The evaluation demonstrates the effectiveness of the feature-enriched approach and demonstrates that features are most useful when data are sparse.

Acknowledgements. This work was in part supported by NSF III-0905117.

References

1. Agarwal, D., Chen, B.-C.: Regression-based latent factor models. In: Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining, pp. 19–28 (2009)

2. Antoniak, C.E.: Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. The Annals of Statistics 2(6), 1152–1174 (1974)
3. Balabanovic, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Commun. ACM 40(3), 66–72 (1997)
4. Blackwell, D., Macqueen, J.B.: Ferguson distributions via Pólya urn schemes. The Annals of Statistics 1, 353–355 (1973)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3(4-5), 993–1022 (2003)
6. Chen, Y.-H., George, E.I.: A bayesian model for collaborative filtering. In: 7th International Workshop on Artificial Intelligence and Statistics (1999)
7. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining, pp. 89–98 (2003)
8. Dunson, D.B., Xue, Y., Carin, L.: The matrix stick-breaking process: Flexible Bayes meta-analysis. Journal of the American Statistical Association 103(481), 317–327 (2008)
9. Ferguson, T.S.: A Bayesian analysis of some nonparametric problems. The Annals of Statistics 1(2), 209–230 (1973)
10. George, T., Merugu, S.: A scalable collaborative filtering framework based on co-clustering. In: Proceedings of the IEEE International Conference on Data Mining, pp. 625–628 (2005)
11. Hartigan, J.A.: Direct clustering of a data matrix. Journal of the American Statistical Association 67(337), 123–129 (1972)
12. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Trans. Inf. Syst. 22, 89–115 (2004)
13. Jacob, L., Hoffmann, B., Stoven, V., Vert, J.-P.: Virtual screening of GPCRs: an in silico chemogenomics approach. BMC Bioinformatics 9(1), 363 (2008)
14. Jin, R., Si, L., Zhai, C.: A study of mixture models for collaborative filtering. Journal of Information Retrieval 9, 357–382 (2006)
15. Khoshneshin, M., Street, W.N.: Incremental collaborative filtering via evolutionary co-clustering. In: Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys 2010, pp. 325–328. ACM, New York (2010)
16. Lemire, D., Maclachlan, A.: Slope one predictors for online rating-based collaborative filtering. In: Proceedings of the SIAM Data Mining, SDM (2005)
17. Lim, Y.J., Teh, Y.W.: Variational Bayesian Approach to Movie Rating Prediction. In: Proceedings of KDD Cup and Workshop (2007)
18. Marlin, B.: Modeling user rating profiles for collaborative filtering. In: Advances in Neural Information Processing Systems (NIPS), vol. 17 (2003)
19. Meeds, E., Roweis, S.: Nonparametric Bayesian Biclustering. Technical Report UTML TR 2007-001, Department of Computer Science, University of Toronto (2007)
20. Neal, R.M.: Markov Chain Sampling Methods for Dirichlet Process Mixture Models. Journal of Computational and Graphical Statistics 9(2), 249–265 (2000)
21. Ning, X., Rangwala, H., Karypis, G.: Multi-assay-based structure activity relationship models: Improving structure activity relationship models by incorporating activity information from related targets. Journal of Chemical Information and Modeling 49(11), 2444–2456 (2009); PMID: 19842624
22. Okuno, Y., Yang, J., Taneishi, K., Yabuuchi, H., Tsujimoto, G.: GLIDA: GPCR-ligand database for chemical genomic drug discovery. Nucleic Acids Research 34(suppl. 1), D673–D677 (2006)

23. Papaspiliopoulos, O., Roberts, G.O.: Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. Biometrika 95(1), 169–186 (2008)
24. Pitman, J., Yor, M.: The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. Annals of Probability 25(2), 855–900 (1997)
25. Porteous, I., Asuncion, A., Welling, M.: Bayesian matrix factorization with side information and dirichlet process mixtures. In: AAAI (2010)
26. Salakhyuditnov, R., Mnih, A.: Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo. In: International Conference on Machine Learning (2008)
27. Schafer, J.B., Konstan, J., Riedi, J.: Recommender systems in e-commerce. In: Proceedings of the ACM Conference on Electronic Commerce, pp. 158–166 (1999)
28. Sethuraman, J.: A constructive definition of Dirichlet priors. Statistica Sinica 4, 639–650 (1994)
29. Shafiei, M., Milios, E.: Latent Dirichlet co-clustering. In: IEEE International Conference on Data Mining, pp. 542–551 (2006)
30. Shan, H., Banerjee, A.: Bayesian co-clustering. In: IEEE International Conference on Data Mining (2008)
31. Shan, H., Banerjee, A.: Generalized probabilistic matrix factorizations for collaborative filtering. In: Proceedings of the IEEE International Conference on Data Mining, pp. 1025–1030 (2010)
32. Sutskever, I., Salakhutdinov, R., Tenenbaum, J.: Modelling relational data using Bayesian clustered tensor factorization. In: Advances in Neural Information Processing Systems, vol. 22, pp. 1821–1828 (2009)
33. Symeonidis, P., Nanopoulos, A., Papadopoulos, A., Manolopoulos, Y.: Nearest-Biclusters Collaborative Filtering with Constant Values. In: Nasraoui, O., Spiliopoulou, M., Srivastava, J., Mobasher, B., Masand, B. (eds.) WebKDD 2006. LNCS (LNAI), vol. 4811, pp. 36–55. Springer, Heidelberg (2007)
34. Wale, N., Karypis, G.: AFGEN. Technical report, Department of Computer Science & Enigneering, University of Minnesota (2007), http://www.cs.umn.edu/~karypis
35. Wang, P., Domeniconi, C., Laskey, K.: Latent Dirichlet Bayesian co-clustering. In: Proceedings of the European Conference on Machine Learning, pp. 522–537 (2009)
36. Xu, Z., Tresp, V., Yu, K., Kriegel, H.: Infinite hidden relational models. In: Proceedings of the International Conference on Uncertainity in Artificial Intelligence (2006)

Shape-Based Clustering for Time Series Data

Warissara Meesrikamolkul, Vit Niennattrakul,
and Chotirat Ann Ratanamahatana

Department of Computer Engineering, Chulalongkorn University
254 Phayathai Road, Pathumwan, Bangkok, Thailand 10330
{g53wms,g49vnn,ann}@cp.eng.chula.ac.th

Abstract. One of the most famous algorithms for time series data clustering is k-means clustering with Euclidean distance as a similarity measure. However, many recent works have shown that Dynamic Time Warping (DTW) distance measure is more suitable for most time series data mining tasks due to its much improved alignment based on shape. Unfortunately, k-means clustering with DTW distance is still not practical since the current averaging functions fail to preserve characteristics of time series data within the cluster. Recently, Shape-based Template Matching Framework (STMF) has been proposed to discover a cluster representative of time series data. However, STMF is very computationally expensive. In this paper, we propose a Shape-based Clustering for Time Series (SCTS) using a novel averaging method called Ranking Shape-based Template Matching Framework (RSTMF), which can average a group of time series effectively but take as much as 400 times less computational time than that of STMF. In addition, our method outperforms other well-known clustering techniques in terms of accuracy and criterion based on known ground truth.

Keywords: Time Series, Clustering, Shape-based Averaging.

1 Introduction

Time series data mining is increasingly an active research area since time series data are ubiquitous, appearing in various domains including medicine [15], geology [13], etc. One of its main mining tasks is clustering, which is a method to seperate unlabeled data into their natural groupings. In many applications related to time series data [14], k-means clustering [2] is generally used with the Euclidean distance function and amplitude averaging (arithmetic mean) as an averaging method.

Although the Euclidean distance is popular and simple, it is not suitable for time series data because its distance between two sequences is calculated in one-to-one manner. As a result, k-means with Euclidean distance does not cluster well because time shifting among data sequences in the same class usually occurs. In time series mining, especially in time series classification, Dynamic Time Warping (DTW) [1] distance has been proved to give more accurate results than Euclidean distance. Unfortunately, k-means clustering with the DTW

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 530–541, 2012.

distance still does not work practically [8][7] because current averaging function does not return a characteristic-preserving averaging result. Traditional k-means clustering fails to return a correct clustering result since this cluster centers do not reflect characteristics of the data, as shown in Fig. 1. In this work, we will demonstrate that our proposed method can resolve this problem.

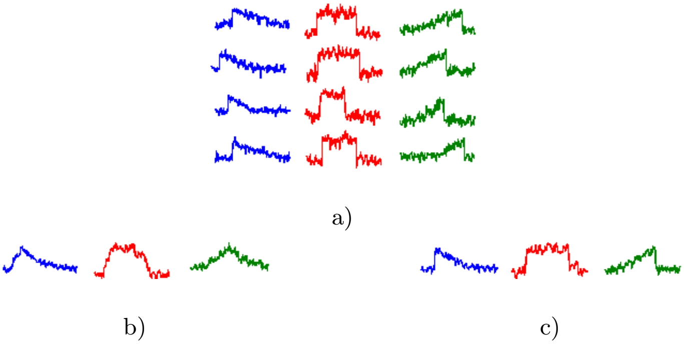


Fig. 1. a) Sample 3-class CBF data [3] and its cluster centers from b) traditional k-means clustering and from c) our proposed method

We propose a novel method called Shape-based Clustering for Time Series (SCTS) which incorporates k-means clustering and DTW distance measure, together with our new averaging method, called Ranking Shape-based Template Matching Framework (RSTMF) extended from Shape-based Template Matching Framework (STMF) [10] for classification. Unlike STMF, our RSTMF uses distances from clustering to approximate an order of sequences to be averaged, giving a few orders of magnitude speedup comparing to STMF. Our evaluation also shows that our proposed method outperforms other well-known clustering techniques in terms of accuracy and criterion based on known ground truth. In addition, the accuracy of our proposed method can future improve when a global constraint [11] is utilized in distance calculation and data averaging.

The rest of the paper is organized as follows. In section 2 and 3, we offer background knowledge and related works. In section 4, we explain our new framework for time series clustering, which is Shape-based Clustering for Time Series (SCTS). The experiments and results are shown in section 5. Finally, conclusions are provided in section 6.

2 Background

This section provides background knowledge on k-means clustering, Dynamic Time Warping (DTW) distance measure, and global constraint.

2.1 *K*-means Clustering

K-means clustering [2] is a well-known and very simple partitioning clustering algorithm. Its algorithm tries to group similar data into the same cluster by using an objective function that minimizes a sum of squared errors between a cluster center to its members. The algorithm is done as follows:

1. Initialize k cluster centers.
2. Measure the similarity between each data and all cluster centers and assign data into the most similar cluster.
3. Calculate a new cluster center of every cluster using an averaging function.
4. Repeat steps 2 and 3 until the cluster membership does not change.

K-means clustering consists of two major subroutines, which are a distance function to measure the similarity between data sequences and an averaging function to return a new cluster center. Generally, most time series clustering works use Euclidean distance and amplitude averaging method. However, both cluster centers and their cluster members are inaccurate. In this work, we resolve this problem by using the DTW distance measure with our newly proposed averaging method called RSTMF.

2.2 Dynamic Time Warping (DTW) Distance Measure

DTW distance [1] is an accurate similarity measurement which is generally used for time series data [9], especially in classification [6]. An optimal alignment and distance between two sequences $P = \langle p_1, \ldots, p_i, \ldots, p_n \rangle$ and $Q = \langle q_1, \ldots, q_j, \ldots, q_m \rangle$ can be determined as follows.

$$DTW(P, Q) = \sqrt{dist(p_n, q_m)} \tag{1}$$

$$dist(p_i, q_j) = (p_i - q_j)^2 + min \begin{cases} dist(p_{i-1}, q_j) \\ dist(p_i, q_{j-1}) \\ dist(p_{i-1}, q_{j-1}) \end{cases} \tag{2}$$

DTW distance is computed through dynamic programming to discover the minimum cumulative distance of each element in $n \times m$ matrix. In addition, the warping path between two sequences can be found by tracing back from the last cell.

In this work, DTW distance is used to measure the similarity between each time series data and cluster centers to give more accurate results.

2.3 Global Constraint

The global constraint is used when we need to limit the amount of warping in the DTW alignment. In some applications such as speech recognition [12], two data sequences are considered the same class when only small time shifting occurs; so,

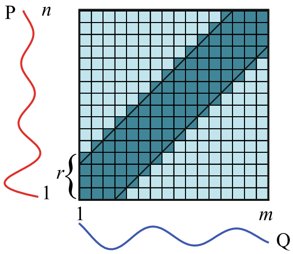


Fig. 2. The warping window of P and Q is limited by the global constraint of size r

the global constraint is used to align the sequences more precisely. The Sakoe-Chiba band [12], one of the most popular global constraints, has been originally proposed for speech community and also has been used in various tasks in time series mining [11]. The size of the warping window is defined by r (as shown in Fig. 2), the percentage of the time series' length, which is symmetric in both above and on the right of a diagonal. In this work, we will show in experiments that the global constraint plays an important role in improving the accuracy.

3 Related Work

In the past few decades, there are many clustering techniques proposed to cluster time series data [5], for example, agglomerative hierarchical clustering [13], which merges most similar objects until all objects are in the cluster. However, this technique is still inaccurate, especially when outliers are present.

Another popular clustering technique is partitional clustering, which tries to minimize an objective function. The well-known algorithms are k-medoids and k-means clustering, which are different in their approaches to find new cluster centers. For k-medoids clustering application [4], DTW distance is used as a similarity measure among data sequences, and a sequence with minimum sum of distance to the rest of the sequences in the cluster is selected as a new cluster center. However, medoid is not always a centroid of a cluster, so the sequences can be assigned to wrong clusters.

In contrast to k-medoids clustering, k-means clustering mostly uses Euclidean distance as a distance metric, and an arithmetic mean or amplitude averaging is simply used to find a new cluster center [14]. Although the DTW distance is more appropriate for time series data, there currently is no DTW averaging method that provides a satisfied averaging result.

According to this, many research works have tried to improve the quality of the averaging result. Shape-based Template Matching Framework (STMF) [10] was recently introduced to average time series sequences. Table 1 shows the algorithm of this framework; the most similar pair of sequences is averaged by Cubic-spline Dynamic Time Warping (CDTW) algorithm (in line 6).

Table 1. Shape-based Template Matching Framework algorithm [10]

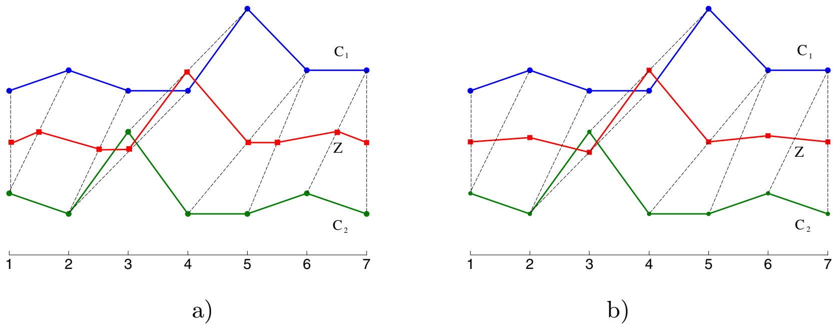
```
Algorithm STMF($D$)
$D$ is the set of time series data to be averaged
initialize weight $\omega = 1$ for every sequences in $D$
while(size($D$) $> 1$)
    $\{C_1, C_2\}$ = the most similar pair of sequences in $D$
    $Z$ = CDTW($C_1$, $C_2$, $\omega_{C_1}$, $\omega_{C_2}$)
    $\omega_Z = \omega_{C_1} + \omega_{C_2}$
    add $Z$ to $D$
    remove $C_1$, $C_2$ from $D$
end while
return $Z$
```

Given C_1 and C_2 as the most similar sequences, first, we find the warping path between these two sequences. The variables c_{1i} and c_{2j} are elements of C_1 and C_2, which are warped. The averaged sequence Z, which has coordinates z_{k_x} and z_{k_y} can be computed as follows.

$$z_{k_x} = \frac{\omega_{c_1} c_{1i} + \omega_{c_2} c_{2j}}{\omega_{c_1} + \omega_{c_2}} \tag{3}$$

$$z_{k_y} = \frac{\omega_{c_1} c_{1i_x} + \omega_{c_2} c_{2j_y}}{\omega_{c_1} + \omega_{c_2}} \tag{4}$$

In equations 3 and 4, ω_{c_1} and ω_{c_2} are the weight of the sequences C_1 and C_2, respectively. After we get the result, a number of points in the averaged sequence is re-sampled by using cubic-spline interpolation [10]. As shown in Fig. 3a), the averaging result from DTW averaging gives a sequence with 9 unequally spaced data points, whereas in Fig. 3b), the sequence is resampled with cubic spline interpolation to obtain a sequence of 7 equally spaced data points.



a) b)

Fig. 3. The average sequences between C_1 and C_2 using DTW alignment a) before applying cubic spline interpolation and b) after applying cubic spline interpolation

However, according to this framework, finding the most similar pair for each time of averaging is enormously computationally expensive because the DTW distance of every pair of the sequences must be computed. Therefore, our RSTMF will mainly focus on improving its time complexity by estimating an order of sequences before averaging while maintaining the accuracy of the averaging results.

4 Shape-Based Clustering for Time Series (SCTS)

In this paper, we propose Shape-based Clustering for Time Series (SCTS) by incorporating k-means clustering and DTW distance, together with a novel averaging function, Ranking Shape-based Template Matching Framework (RSTMF). Although STMF can still be used to determine a cluster center, it is computationally expensive; therefore, computational time of k-means clustering significantly increase.

We provide an overview of the proposed clustering algorithm in Table 2; the DTW distance is used instead of the Euclidean distance in a membership assignment process. After we finished assigning each data sequence into the most similar cluster, RSTMF is utilized to average all of the sequences within each cluster until all cluster centers are updated. Unlike STMF, RSTMF approximates an order of averaged sequences by looking at the $Dist$ value, which is the DTW distance between data sequences in M and all cluster centers in C. Accordingly, RSTMF can provide the average sequence by using less computation time than that of STMF, which calculates the distance between every pair of data and the most similar pair of sequences is averaged, making it very computationally expensive.

Table 3 shows our RSTMF averaging algorithm, which determines a cluster center by using Cubic-spline Dynamic Time Warping (CDTW) [10] to average a pair of time series sequences. RSTMF utilizes $Dist$ to approximate a similarity distance between every sequence pair, defined by $dist_{approx}$. After that, CDTW is used to average a pair of sequences with the minimum $dist_{approx}$ value. Then, we update S and continue the averaging until only one sequence remains.

In RSTMF algorithm, the $dist_{approx}$ between each pair of the sequences can be computed by using the $Dist$ value. Suppose P and Q are data sequences in M, we have $Dist_{M_P,\ldots} = \langle Dist_{M_P,C_1}, \ldots, Dist_{M_P,C_k}, \ldots, Dist_{M_P,C_K} \rangle$ and $Dist_{M_Q,\ldots} = \left\langle Dist_{M_Q,C_1}, \ldots, Dist_{M_Q,C_k}, \ldots, Dist_{M_Q,C_K} \right\rangle$ where $Dist_{M_P,C_k}$ and $Dist_{M_Q,C_k}$ are the distance between P or Q and its k^{th} cluster center, and K is a number of cluster. By applying the triangular inequality theorem, p_k and q_k are assumed to be two sides of a triangle. Then, the $dist_{approx}$ of P and Q, which is another side of the triangle, can be approximated by equation 5 and collected into S.

$$dist_{approx}(Dist_{M_P,\ldots}, Dist_{M_Q,\ldots}) = \max_{1 \le k \le K} \left| Dist_{M_P,C_k} - Dist_{M_Q,C_k} \right| \qquad (5)$$

After finishing an averaging of two sequences, we insert the resulting sequence into M and delete these two sequences. Then, we update S by using the algorithm in Table 4.

Table 2. Shape-based Clustering for Time Series (SCTS)

```
**Algorithm** SCTS($D$, $K$)
$D$ is the set of time series data
$C$ is the set of cluster centers
$K$ is the number of cluster in $C$
$M$ is the set of data in each cluster
$Dist$ is the matrix of the distance between data sequences and all cluster centers
initialize $C$ as cluster centers of $K$ clusters
do
  for $i$ = 1:size($D$)
    for $k$ = 1:$K$
      $Dist_{D_i,C_k}$ = DTW($D_i$, $C_k$)
    end for
    if($Dist_{D_i,C_k}$ is minimal)
      assign $D_i$ into $M_k$
    end if
  end for
  for $k$ = 1:$K$
    $C_k$ = RSTMF($M_k$, $Dist$)
  end for
while(the cluster membership changes)
return the cluster members and the cluster centers
```

Table 3. The RSTMF algorithm

```
**Algorithm** RSTMF($M$, $Dist$)
$M$ is the set of data in each cluster
$Dist$ is the matrix of the distance between data sequences and all cluster centers
$S$ is the matrix of the distance between data sequences in $M$
initialize weight $\omega = 1$ for every sequences in $M$
for $i$ = 1:size($M$)
  for $j$ = $i$+1:size($M$)
    $S_{M_i,C_j} = S_{M_j,C_i} = dist_{approx}(Dist_{M_i,\ldots}, Dist_{M_j,\ldots})$
  end for
end for
while(size($M$) > 1)
  $S_{M_i,C_j}$ = minimum value in $S$
  $M_z$ = CDTW($M_i$, $M_j$, $\omega_{M_i}$, $\omega_{M_j}$)
  $\omega_{M_z} = \omega_{M_i} + \omega_{M_j}$
  add $M_z$ to $M$
  UPDATE($S$, $i$, $j$, $z$)
  remove $M_i$, $M_j$ from $M$
end while
return $M_z$
```

Table 4. The UPDATE algorithm

```
Algorithm UPDATE($S$, $a$, $b$, $z$)
$S$ is the matrix of the distance between data sequences in $M$
for $i$ = 1:size($S$)
    $S_{M_z,M_i} = S_{M_i,M_z} = \min(S_{M_a,M_i}, S_{M_b,M_i})$
end for
remove $S_{M_a,\ldots}$, $S_{\ldots,M_a}$, $S_{M_b,\ldots}$, $S_{\ldots,M_b}$ from $S$
```

By using the $dist_{approx}$ and the UPDATE method, our RSTMF can achieve large speedup because we can estimate an order of the sequences before averaging. In contrast, the original STMF needs to calculate the DTW distance to select the most similar pair of the sequences every time of averaging.

5 Experiments and Results

In this work, we evaluate our method by comparing it with other clustering techniques, which are typical k-means clustering with the Euclidean distance and amplitude averaging function, k-medoids clustering with the DTW distance [4], and k-hierarchical clustering [13] using both the Euclidean and the DTW distance. We compare our SCTS using RSTMF with that using the original STMF. Our experiments are evaluated on ten datasets from the UCR datasets classification/clustering archive [3] in diverse domains, as shown in Table 5.

Table 5. The details of datasets

Datasets	Number of classes	Length of data	Size of training set	Size of test set
Synthetic Control	6	60	300	300
Trace	4	275	100	100
Gunpoint	2	150	50	150
Lightning-2	2	637	60	61
Lightning-7	7	319	70	73
ECG	2	96	100	100
Olive Oil	4	570	30	30
Fish	7	463	175	175
CBF	3	128	30	900
Face Four	4	350	24	88

We execute each algorithm for 40 times with random initial cluster centers, and the k value is set to the a number of classes in each dataset. With the luxury of labeled datasets used in all experiments, an accuracy, which is the number of correctly assigned data sequences in all clusters, is used evaluation. Fig. 4 shows the accuracy of our proposed method, comparing other well-known clustering methods mentioned above. According to the results, our method outperforms others in almost all datasets.

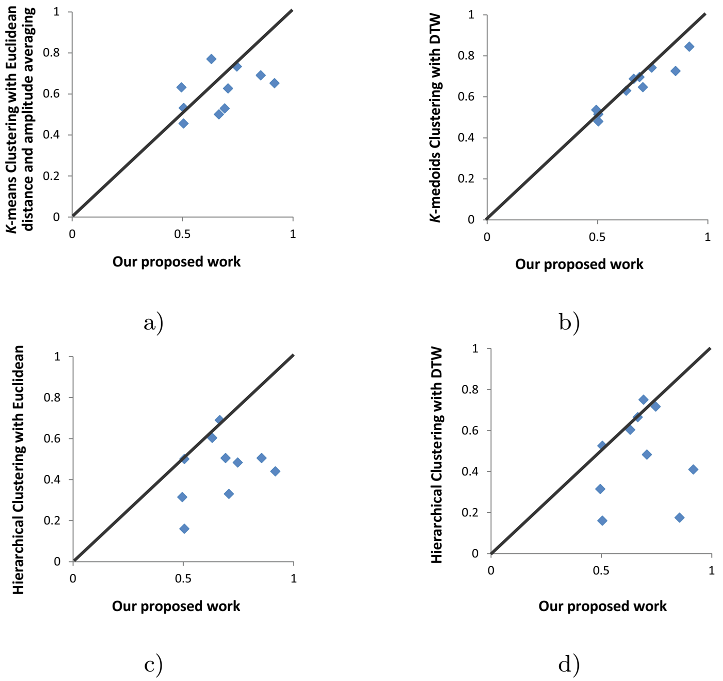


Fig. 4. The accuracy of our RSTMF method on 10 datasets, comparing with a) general k-means clustering, b) k-medoids clustering, and k-hierarchical clustering using c) the Euclidean distance and d) the DTW distance, respectively

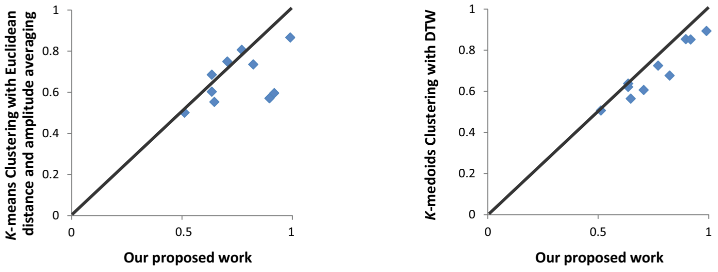
To re-emphasize our finding, we also use another criterion based on known ground truth [5] to measure a similarily between two sets of clusters, i.e., ground-truth clusters and results from clustering algorithms. Suppose G and C are sets of k ground truth clusters and the clusters from our clustering technique. The similarity between G and C is calculated by the following equations.

$$Sim(G,C) = \frac{1}{k}\sum_{i=1}^{k} \max_{1\leq j\leq k} Sim(G_i, C_j) \tag{6}$$

$$Sim(G_i, C_j) = \frac{2\,|G_i \cap C_j|}{|G_i| + |C_j|} \tag{7}$$

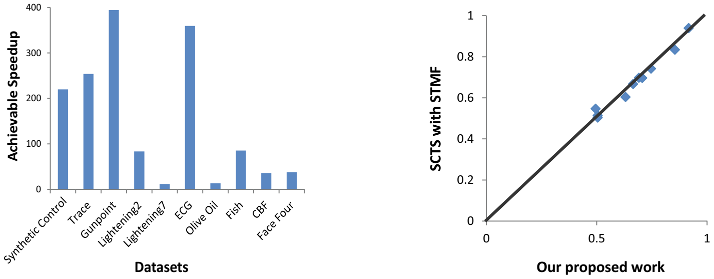
In Fig. 5, we compare our proposed work with the general k-means clustering and the k-medoids clustering using this criterion. The results show that the clusters obtained from our method are more similar to the ground-truth clusters because the RSTMF averaging method does give the new cluster centers that represent the overall characteristic of the data within each cluster.

Furthermore, RSTMF can reduce the time complexity by a few orders of magnitude (as shown in Fig. 6a), while still providing comparable accuracy to STMF (as shown in Fig. 6b).



a) b)

Fig. 5. The criterion based on known ground truth, comparing our proposed method with a) general k-means clustering and b) k-medoids clustering



a) b)

Fig. 6. a) The speedup achieved by our proposed work. b) The accuracy of our proposed work comparing with that using STMF.

In some cases, it appears that SCTS with DTW distance achieves a lower accuracy than the general k-means clustering. In an attempt to alleviate this drawback, we experiment on the global constraint parameter of DTW, Sakoe-Chiba band. We can improve the clustering accuracy, comparing with the original k-means clustering (warping window size is 0%). Fig. 7 shows the accuracy of our proposed RSTMF and STMF, which are comparable, as warping window sizes vary. In almost datasets, the larger warping window size does not always provide the better accuracy; so, the appropriate warping window size is around 20%. However, in some dataset such as ECG, the wider warping window can lead to pathological warping and make the accuracy of clustering decreases.

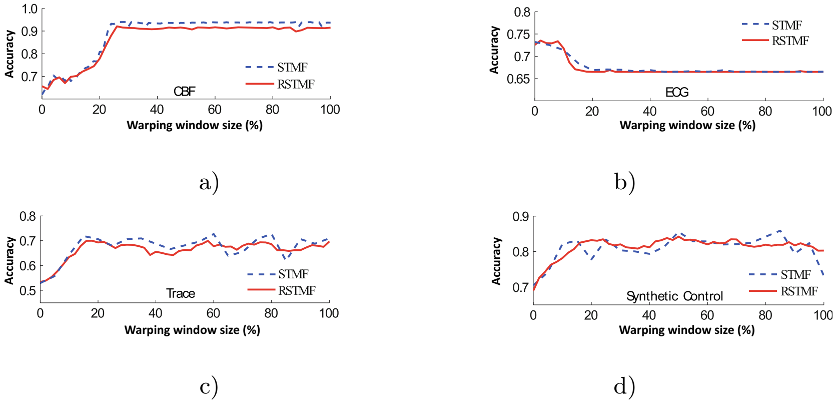


Fig. 7. The accuracy of Shape-based clustering using STMF and our proposed RSTMF of a) CBF, b) ECG, c) Trace, and d) Synthetic Control datasets

6 Conclusion

In this paper, we propose time series data clustering technique called Shape-based Clustering for Time Series (SCTS), which incorporates k-means clustering with a novel averaging method called Ranking Shape-based Template Matching Framework (RSTMF).

Comparing with the other well-known clustering algorithms, our SCTS yields better cluster results in terms of both accuracy and the criterion based on known ground truth because our RSTMF averaging function provides cluster centers that preserve characteristics of data sequences within the cluster (as shown in Fig. 8). Furthermore, RSTMF does gives a comparable sequence averaging result while consuming much less computational time than STMF in a few orders of magnitude; therefore, RSTMF is practically applied in clustering algorithm. We also used global constraint to increase an accuracy of our clusters. The results show that our SCTS can provide more accurate clustering when the width of warping window is about 20% of time series length.

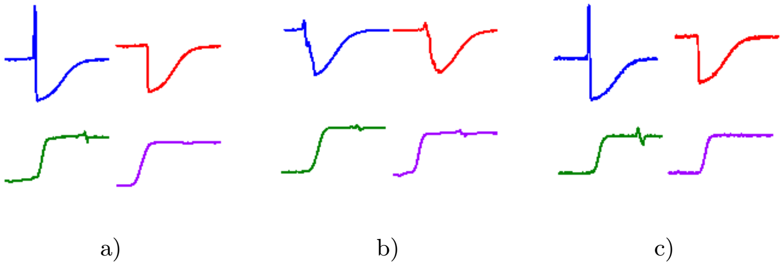


Fig. 8. The cluster centers obtained from a) our proposed method and b) the original k-means clustering of c) sample 4-class Trace data

Acknowledgements. This research is partially supported by the Thailand Research Fund (Grant No. MRG5380130), the Thailand Research Fund given through the Royal Golden Jubilee Ph.D. Program (PHD/0141/2549 to V. Niennattrakul and C.A. Ratanamahatana), and CU Graduate School Thesis Grant.

References

1. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: Proceedings of AAAI Workshop on Knowledge Discovery in Databases, pp. 359–370 (1994)
2. Bradley, P.S., Fayyad, U.M.: Refining initial points for k-means clustering. In: Proceedings of the International Conference on Machine Learning (ICML 1998), pp. 91–99 (1998)
3. Keogh, E., Xi, X., Wei, L., Ratanamahatana, C.A. (2011), http://www.cs.ucr.edu/~eamonn/time_series_data
4. Liao, T.W., Bodt, B., Forester, J., Hansen, C., Heilman, E., Kaste, R.C., O'May, J.: Understanding and projecting battle states. In: Proceedings of 23rd Army Science Conference (2002)
5. Liao, T.W.: Clustering of time series data-a survey. Pattern Recognition, 1857–1874 (2005)
6. Meesrikamolkul, W., Niennattrakul, V., Ratanamahatana, C.A.: Multiple shape-based template matching for time series data. In: Proceedings of the 8th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2011), pp. 464–467 (2011)
7. Niennattrakul, V., Ratanamahatana, C.: On clustering multimedia time series data using k-means and dynamic time warping. In: Proceedings of the International Conference on Multimedia and Ubiquitous Engineering, pp. 733–738 (2007)
8. Niennattrakul, V., Ratanamahatana, C.A.: Inaccuracies of Shape Averaging Method Using Dynamic Time Warping for Time Series Data. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2007. LNCS, vol. 4487, pp. 513–520. Springer, Heidelberg (2007)
9. Niennattrakul, V., Ruengronghirunya, P., Ratanamahatana, C.: Exact indexing for massive time series databases under time warping distance. Data Mining and Knowledge Discovery 21, 509–541 (2010)
10. Niennattrakul, V., Srisai, D., Ratanamahatana, C.A.: Shape-based template matching for time series data. Knowledge-Based Systems 26, 1–8 (2011)
11. Ratanamahatana, C.A., Keogh, E.: Making time-series classification more accurate using learned constraints. In: Proceedings of SIAM International Conference on Data Mining (SDM 2004), pp. 11–22 (2004)
12. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech and Signal Processing, 43–49 (1978)
13. Shumway, R.H.: Time-frequency clustering and discriminant analysis. Statistics and Probability Letters, 307–314 (2003)
14. Vlachos, M., Lin, J., Keogh, E., Gunopulos, D.: A wavelet-based anytime algorithm for k-means clustering of time series. In: Proceedings of Workshop on Clustering High Dimensionality Data and Its Applications, pp. 23–30 (2003)
15. Wismuller, A., Lange, O., Dersch, D.R., Leinsinger, G.L., Hahn, K., Pütz, B., Auer, D.: Cluster analysis of biomedical image time-series. International Journal of Computer Vision, 103–128 (2002)

Privacy-Preserving EM Algorithm for Clustering on Social Network

Bin Yang[1], Issei Sato[2], and Hiroshi Nakagawa[2]

[1] Graduate School of Information Science and Technology, The University of Tokyo
{yangbin,sato}@r.dl.itc.u-tokyo.ac.jp
[2] Information Technology Center, The University of Tokyo
nakagawa@dl.itc.u-tokyo.ac.jp

Abstract. We consider the clustering problem in a private social network, in which all vertices are independent and private, and each of them knows nothing about vertices other than itself and its neighbors. Many clustering methods for networks have recently been proposed. Some of these works have dealt with a mixed network of assortative and disassortative models. These methods have been based on the fact that the entire structure of the network is observable. However, entities in real social network may be private and thus cannot be observed. We propose a privacy-preserving EM algorithm for clustering on distributed networks that not only deals with the mixture of assortative and disassortative models but also protects the privacy of each vertex in the network. In our solution, each vertex is treated as an independent private party, and the problem becomes an n-party privacy-preserving clustering, where n is the number of vertices in the network. Our algorithm does not reveal any intermediate information through its execution. The total running time is only related to the number of clusters and the maximum degree of the network but this is nearly independent of the total vertex number.

1 Introduction

The analysis of social networks has attracted increasing amounts of attention in recent years since there have been progressively more social applications used in practice. Many clustering algorithms with respect to vertices in networks, on the other hand, such as the graph min-cut and label propagation, have been proposed. Most of these methods deal with a so-called assortative mixing model, in which vertices are divided into groups such that the members of each group are mostly connected to other members of the same group [1]. For example, in a communication network (Fig. 1(a)), each student belongs to either of two clubs. They communicate with other students in the same club more frequently than they do with students outside the club. Thus, the methods used for the assortative mixing model could be used to detect the structures in this kind of network. Inversely, in the disassortative mixing networks, vertices have most of their connections outside their group. For example, there are two groups of people in Fig. 1(b), producers and consumers. Most of their exchanges, denoted by the edges,

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 542–553, 2012.

will occur between these two classes. Even though both assortative and disassortative mixing models have theoretical and practical significance, their mixture is more meaningful in most practical applications. If, for example (Fig. 1(c)), researchers in the same field were treated as one group, there would generally be more connections inside each group. In addition, some cross-disciplinary researchers, such as researchers in computational linguistics, may regularly connect with researchers in other related fields, such as linguistics, psychology, and computer science, although they may frequently also connect with other members of the same group. Newman et al. [1] proposed a probabilistic mixture model that could deal with an assortative and disassortative mixture using an EM algorithm. Such a model is realistic for the clustering problem in social networks.

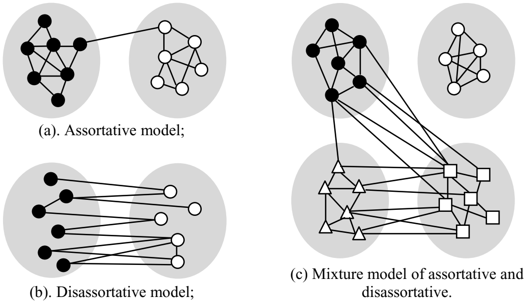

(a). Assortative model;
(b). Disassortative model;
(c) Mixture model of assortative and disassortative.

Fig. 1. *Assortative* model, *disassortative* model and mixture model

With increasing concerns about the issue of personal information and privacy protection, many privacy-preserving data mining algorithms have been proposed. In this paper, we consider the clustering problem on social networks, in which each member contacts the others via various means of communication, such as social applications (MSN, Yahoo Messenger, etc.), mobile phones of different service providers, etc. Their records are stored in different organizations, such as Microsoft, Yahoo, and mobile service providers. The collection of their data always contains large commercial value. However, it is impossible to make these competitors collaborate to perform data mining algorithms, such as clustering. This motivate us to develop a secure clustering algorithm that can be performed without any support of the organizations. Using this algorithm, not only vertices in the network are clustered, but the privacy of each vertex is also protected.

We summarize the related works in section 2. Section 3 introduces some background knowledge and Section 4 formulates our problem. We develop two basic secure summation protocols in Section 5, and propose our main EM-algorithm for private clustering based on these protocols in Sections 6. In Section 7, we discuss our evaluation of the performance of our protocols. The results of experiments conducted to evaluate the performance of our protocols are explained in Section 8, and concluding remarks are given in the last section.

2 Related Work

Newman et al. [1] proposed a probabilistic model for the mixture of assortative and disassortative models, and provided a corresponding EM algorithm, by which all vertices are clustered so that vertices in the same cluster had the same probabilities as if they had a connection with each vertex in the network.

Many kinds of privacy-preserving methods have been proposed to carry out data mining while protecting privacy. In general, privacy-preserving K-means clustering problems can be classified into horizontally partitioned K-means [4], vertically partitioned K-means [5] and arbitrarily partitioned K-means [6]. Methods using privacy-preserving EM clustering have also been proposed [7]. All of these methods deal with distributed databases with large numbers of data.

Secure data analysis in networks has recently attracted increasingly more attention. Hay et al. [8] proposed an efficient algorithm to compute the distribution of degrees of social networks. Another method of computing users privacy scores in online social networks was provided by Liu et al. [9]. Sakuma et al. [10] used the power method to solve ranking problems such as PageRank and HITS where each vertex in a network was treated as one party and only knew about its neighbors. Similar work was done by Kempe et al. [11].

Even though all these works provided valuable studies, clustering in private peer-to-peer networks has not yet attracted adequate attention. We focus on the clustering problem based on these kinds of private networks. In addition, we also concentrate on the mixture of assortative and disassortative models.

3 Preliminaries

Let us consider an un-weighted directed network of n vertices, numbered $1, \cdots, n$. The adjacency matrix of the network is denoted by $\mathbf{A}$ with elements $A_{ij} = 1$, if there is an edge from i to j, and 0 otherwise. If there is an adjacency from i to j, we say that i is a parent of j, and j is a child of i. We denote $pa(i)$ as the set of all parents of vertex i and $ch(i)$ as the set of its children. The union of $pa(i)$ and $ch(i)$ is referred to as the neighbors of i. All vertices fall into C clusters, and g_i denotes the cluster to which vertex i belongs. These g_is are treated as unknown or hidden data, and the purpose of our model is to deduce g_is from the adjacency matrix $\mathbf{A}$. We use the notation $[C]$ to denote the collection, $\{1, 2, \cdots, C\}$.

3.1 Probabilistic Mixture Model

Let θ_{ri} denote the probability that a link from a vertex in cluster r is connected to vertex i, and π_r denote the fraction of vertices in cluster r. The normalization conditions ($\sum_{r=1}^{C} \pi_r = 1$, $\sum_{i=1}^{n} \theta_{ri} = 1$) are satisfied. Using the probabilistic mixture model [1], the structural features in large-scale network can be detected by dividing the vertices of a network into clusters, such that the members of each cluster had similar patterns of connections to other vertices. We illustrate this generative graphical model in Fig. 2.

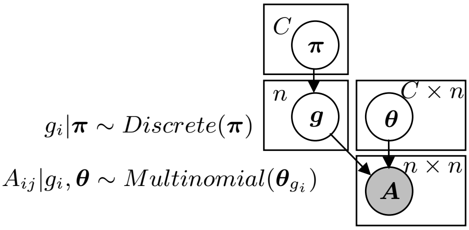

C
π
n
g
θ
C × n
n × n
A
$g_i|\boldsymbol{\pi} \sim Discrete(\boldsymbol{\pi})$
$A_{ij}|g_i, \boldsymbol{\theta} \sim Multinomial(\boldsymbol{\theta}_{g_i})$

Fig. 2. Probabilistic generative model of mixture network

In Fig. 2, $\boldsymbol{\pi}$ expresses the vector $(\pi_1, \pi_2, \cdots, \pi_C)$; $\boldsymbol{\theta}_r$ expresses the vector $(\theta_{r1}, \theta_{r2}, \cdots, \theta_{rn})$ and $\boldsymbol{\theta}$ expresses the matrix $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_C)^T$. In this model, hidden variables g_is (the cluster labels of vertices) are generated from a discrete probability distribution with parameter $\boldsymbol{\pi}$, i.e., $\Pr(g_i = k) = \pi_k$, which means the probability that vertex i belongs to cluster k is π_k. After all g_is are determined, each vertex will choose some vertices with a multinomial distribution with corresponding parameter $\boldsymbol{\theta}_{g_i}$ (the g_i^{th} row in $\boldsymbol{\theta}$), and connect itself to each chosen vertex. Since the members of each cluster share the same parameter $\boldsymbol{\theta}_{g_i}$, they have similar patterns of connections to other vertices in the network.

Newman et al. [1] also proposed an EM-algorithm to infer the probabilities of these hidden variables. The E-step and M-step are derived as follows.

$$\text{E-step:} \quad q_{ir} = \frac{\pi_r \prod_{j=1}^{n} \theta_{rj}^{A_{ij}}}{\sum_{s=1}^{C} \pi_s \prod_{j=1}^{n} \theta_{sj}^{A_{ij}}}; \tag{1}$$

$$\text{M-step:} \quad \pi_r = \frac{1}{n} \sum_{i=1}^{n} q_{ir}; \quad \theta_{rj} = \frac{\sum_{i=1}^{n} A_{ij} q_{ir}}{\sum_{i=1}^{n} (\sum_{j=1}^{n} A_{ij}) q_{ir}}. \tag{2}$$

Here q_{ir} is defined as the probability that vertex i is a member of cluster r:

$$q_{ir} = \Pr(g_i = r | A, \pi, \theta) \ . \tag{3}$$

Since q_{ir}s denote the probability of g_i, inferring q_{ir}s is equivalent to inferring g_i. The adjacency matrix, $\mathbf{A}$, is treated as observed data. q_{ir}s are initialized to be arbitrary values in the beginning of the EM-algorithm, and converge to the final results after several rounds of E-steps and M-steps.

3.2 Utilities of Privacy-Preserving Data Mining

Homomorphic Encryption. In a public key encryption system, a public key pk, used to encrypt a given message, and a private key sk, used to decrypt the cryptograph, are generated by an asymmetric key algorithm. The private key is kept secret, while the public key may be widely distributed. Given a plaintext m, $c = E_{pk}(m, t)$ denotes a random encryption of m, and $d = D_{sk}(c)$ denotes the decryption of c, where t is randomly generated from Z_N, and N is a large positive integer. Paillier encryption [12] is a public key encryption system. It also satisfies additive homomorphism, i.e., there is an operation "$\cdot$", s.t. $\forall m_1, m_2 \in Z_N$,

$$E_{pk}(m_1 + m_2, t) \equiv E_{pk}(m_1, t_1) \cdot E_{pk}(m_2, t_2) \pmod{N^2} , \tag{4}$$

where t, t_1 and t_2 are random numbers. We will omit these random numbers when they were not necessary. Using this property, we can securely compute the cryptograph of the summation of two numbers, m_1 and m_2, only given their cryptographs. The following condition can be obtained from (4).

$$E_{pk}(m \cdot k) \equiv E_{pk}(m)^k \pmod{N^2} . \tag{5}$$

Secure Summation Protocols. Suppose each party has a private input. All the parties collaborate to compute the summation of all their inputs, without any party obtaining any information about other parties. Such a protocol is called a secure summation protocol. Many secure summation protocols, such as those by Kantarcoglu et al. [13], have been proposed. But these methods have been based on the assumption that any two parties are connected.

4 Problem Statement

We focus on the clustering problem in a social network described in Section 3. Furthermore, we also protect the privacy of each vertex in the network.

4.1 Assumptions

We treat each vertex in the network as one party. Thus, a network containing n vertices becomes an n-party system. We also assume that this network is a connected network, in which there is at least one path between any pair of vertices. There is no special vertex in the network, i.e., each vertex performs the same operations. These assumptions are of practical significance. For example, the relations of sending e-mail can be used to construct a network. Each vertex (an e-mail user) can be seen as one party. Hence, each e-mail user knows its neighbors, since it is connected to each neighbor using e-mail. Also, all e-mail users in this network are equivalent.

Since the vertices, such as e-mail users, never want to reveal private information about themselves in practice, we need to consider the privacy of each vertex in the network. We specifically assume that each vertex only knows about itself and its neighbors. First, it knows all information about itself. Second, it only knows about the connections with its neighbors. Third, it knows nothing about other vertices, not even whether they exist. Moreover, we assume all parties are semi-honest, which means that they all correctly follow the protocol with the exception that they keep a record of all their intermediate computations.

The knowledge range of any vertex is outlined in Fig. 3. We take the white vertex as the current vertex. It only knows about its neighbors (gray) since there is an edge between them. However, it does not know anything about the other (black) vertices, and even does not know whether they exist. In addition, it even does not know whether any pair of its neighbors is connected or not.

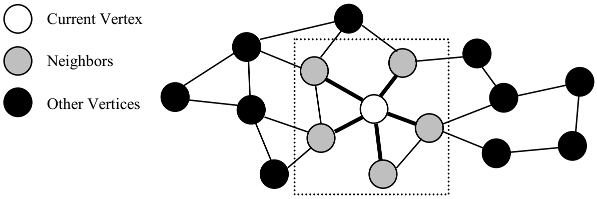


Fig. 3. Range of vision of a vertex in private network

Table 1. Protocol 1 - Local Secure Summation Protocol

```
Inputs: Party $i$ has an $x_i$, for $i = 0, 1, 2, \cdots, m$;
Outputs: Party 0 gets $x = \sum_{i=0}^{m} x_i$; other parties get nothing;
Party $m$ generates a set of keys $(pk, sk)$;
  $pk$ is published to all parties; $sk$ is known to party $m$ only;
For $i = 1$ to $m - 1$
    Party $i$ encrypts its input: $X_i = E_{pk}(x_i)$, and sends $X_i$ to Party 0;
Party 0 generates a random number $r_0$, and encrypts it: $R_0 = E_{pk}(r_0)$;
Party 0 computes $Z = R_0 \cdot \prod_{i=1}^{m-1} X_i$, and sends $Z$ to Party $m$;
Party $m$ decrypts $Z$: $z = D_{sk}(Z) = r_0 + \sum_{i=1}^{m-1} x_i$;
Party $m$ computes $z' = z + x_m$, and send $z'$ to Party 0;
Party 0 computes $x = z' - r_0 + x_0 = \sum_{i=0}^{m} x_i$.
```

4.2 Private Variables and Public Variables

In our private network, the variables A_{ij}, π_r, θ_{rj}, and q_{ir} are distributed into all parties. The A_{ij} denotes whether the pair of (i, j) is connected, so we treat it as private information of parties i and j. The π_r denotes the fraction of vertices in cluster r. As it contains nothing about individual parties, we publish it to all parties. The θ_{rj} expresses the relationship between cluster r and vertex j. We assume it is only known to party j. The q_{ir} is similarly only known to party i.

5 Secure Summation Protocols on Networks

We propose two secure summation protocols for the private network.

5.1 Local Secure Summation Protocol

Suppose a party, numbered 0, has m children, numbered $1, 2, \cdots, m$, in turn. Each of these parties has a private input x_i $(i = 0, 1, \cdots, m)$. After this protocol is executed, party 0 securely obtains the summation: $x = \sum_{i=0}^{m} x_i$, while other parties obtain nothing. We assume that party 0 only communicates with its children, and these children do not communicate with each other.

The detail of this protocol is shown in Table 1. Since only Party m can decrypt messages, Party 0 can obtain nothing about x_i from X_i $(i \in \{1, 2, \cdots, m-1\})$. From the homomorphism of encryption, $Z = E_{pk}(r_0 + \sum_{i=1}^{m-1} x_i)$, Party m can then compute $z = r_0 + \sum_{i=1}^{m-1} x_i$. As Party m does not know r_0, it can obtain nothing about $\sum_{i=1}^{m-1} x_i$ from the values of z. In summary, nothing can be inferred from the intermediate information other than the final result, x.

5.2 Global Secure Summation Protocol

The goal of this protocol is to securely sum up the inputs of all parties in our distributed network without revealing the privacy of any party. The final result is published to all parties throughout the network. Under the assumption in Section 4, each party can only communicate with its neighbors. Nevertheless, we can arbitrarily choose one party as a root and construct a spanning tree $\mathbf{T}$ from the network, since the network is connected. We use $\mathbf{T}$ to accumulate and distribute data.

Table 2. Protocol 2 - Global Secure Summation Protocol

```
Inputs: Party $i$ has an $x_i$, where $i = 1, 2, \cdots, n$, (all vertices in the network)
        where 1 is the root, and $2, 3, \cdots, m$ are all children of root in $\mathbf{T}$;
Outputs: All parties get the summation $x = \sum_{i=1}^{n} x_i$;
01  Choose a leaf party, called $n$, who has no child in $\mathbf{T}$;
02  Party $n$ generates a set of keys $(pk, sk)$;
03    $pk$ is published to all parties via the paths in $\mathbf{T}$; $sk$ is known to party $n$ only;
04  Each party encrypts its input: $X_i = E_{pk}(x_i)$;
05  Each party $i$ computes $Y_i$ as the following
06    $Y_i := X_i \cdot \prod_j Y_j$ ($j$ is the child of $i$ in $\mathbf{T}$);
07  Party 1 sends $Y_1$ to Party $n$ via the paths in $\mathbf{T}$;
08  Party $n$ decrypts $Y_1$: $x = D_{sk}(Y_1) = \sum_{i=1}^{n} x_i$;
09  Party $n$ publishes $x$ to all parties via the paths in $\mathbf{T}$.
```

The detail of this protocol is shown in Table 2. The equation in line 06 implies that each vertex accumulates the cryptographs of the summation of the sub-tree of itself, and sends the result, Y_i, to its parent in $\mathbf{T}$. Hence Y_1 in the root is the cryptograph of the summation of all vertices. Moreover, since only Party n can decrypt messages, nothing is revealed through the execution of the protocol other than the final result.

6 Private Clustering on Networks

The main procedure in our private clustering is the same as the original method, in which E-step and M-step were performed repeatedly until convergence. The only difference is that we need to protect the privacy of each vertex in each step.

6.1 Private E-step

In the private E-step, each Party i computes its private q_{ir}s in (1) without revealing θ_{rj}s. We now simplify the E-step (1) by introducing a new variable:

$$\alpha_{ir} = \pi_r \prod_{j=1}^{n} \theta_{rj}^{A_{ij}} \ . \tag{6}$$

Hence, the q_{ir} of party i can be rewritten as follows, for $r \in [C]$.

$$q_{ir} = \frac{\alpha_{ir}}{\sum_{s=1}^{C} \alpha_{is}} \ . \tag{7}$$

If party i obtains the values of α_{ir}s ($r \in [C]$), the q_{ir}s can be directly computed. Consequently, we focus on securely computing of α_{ir}s (6). Although (6) is a product of n items, from the definitions of A_{ij}, we could eliminate the term θ_{rj} if party j is not a child of party i. In other words, the value of α_{ir} becomes the product of π_r and the θ_{rj}s of all children of party i, i.e.,

$$\alpha_{ir} = \pi_r \cdot \prod_{A_{ij}=1} \theta_{rj} = \pi_r \cdot \prod_{j \in ch(i)} \theta_{rj} \ . \tag{8}$$

Hence, we have

$$\log \alpha_{ir} = \log \pi_r + \sum_{j \in ch(i)} \log \theta_{rj} \ . \tag{9}$$

Here, each $\log \theta_{rj}$ can be seen as a private input of party j. Then, our goal becomes to securely compute the summation of these $\log \theta_{rj}$s ($j \in ch(i)$). To do this, we only need to perform Protocol 1 by treating these $\log \theta_{rj}$s ($j \in ch(i)$) as the parameters of this protocol (private inputs of children of party i). Throughout this execution, the value of each θ_{rj} is kept secret with each party.

6.2 Private M-step

In the private M-step, each Party j computes θ_{rj}s and all parties obtain the π_{r}s in (2) without revealing any q_{ir}s. We now introduce a new variable:

$$\beta_{rj} = \sum_{i=1}^{n} A_{ij} q_{ir}, \quad \beta_r = \sum_{j=1}^{n} \beta_{rj} \ . \tag{10}$$

Similarly to (8), β_{rj} can be rewritten as:

$$\beta_{rj} = \sum_{A_{ij}=1} q_{ir} = \sum_{i \in pa(j)} q_{ir} \ . \tag{11}$$

Similarly, treating q_{ir}s as the private inputs of its parents, party j can securely compute the value of β_{rj} with Protocol 1 without revealing any information about q_{ir}s. In addition, substituting the definition of β_{rj} into (2), θ_{rj} becomes

$$\theta_{rj} = \frac{\beta_{rj}}{\sum_{k=1}^{n} \beta_{rk}} = \frac{\beta_{rj}}{\beta_r} . \tag{12}$$

Since Party j does not know the values of β_{rk}s for $k \neq j$, it cannot compute the $\beta_r = \sum_{k=1}^{n} \beta_{rk}$. As β_r does not include any private information, we publish β_r ($r \in [C]$) to all parties. The problem of computing β_r becomes that of securely computing the summation of the private inputs of all parties in the network and publishing the final result to all parties. Hence, it can be solved with Protocol 2. Given the values of β_rs ($r \in [C]$), party j can compute θ_{rj}s using (12) by itself.

Using the definition of π_r and treating q_{ir} as the private input of each party in the network, securely computing π_r is equivalent to securely summing all parties in the entire network and it can thus be solved using Protocol 2.

7 Performance

We discuss the efficiency of our method here. Both computation and communication can be carried out in parallel in the execution of our protocol. As each party performs operators with only one neighbor at the same time, evaluating the total running time is equivalent to the edge coloring problem in graph theory. The edge coloring of a graph is generally the assignment of "*colors*" to its edges so that no two adjacent edges have the same color. Vizing [15] has shown that the color index of a graph with maximum vertex-degree K is either K or $K+1$.

We now discuss the running time for one round of computation, which includes one E-step and one M-step. In the E-step, each party i performs Protocol 1 with all its children for C times. From Vizings conclusion [15], the total running time for this stage is $O(CK)$. In the M-step, the β_rs and π_rs are all accumulated with Protocol 2. Because the running time for one duration of Protocol 2 is $O(\log_K n)$ and the β_rs and π_rs include $2C$ values, the running time for these accumulations is $O(C \log_K n)$. The secure computation of β_{rj} involves C times of executions of Protocol 1 with all its parents. From Vizings conclusion [15], the total running time for this computation is at most $O(CK)$.

In summary, the running time for one round of E-step and M-step is $O(CK + C \log_K n)$. Nevertheless, this is just an atomic operation of the entire EM-algorithm. If we need to perform R rounds of E-step and M-step until they converge, the entire running time will become $O(RC(K + \log_K n))$.

8 Experiments

We implemented the protocols in C++ using the OpenSSL library, which is an implementation with large numbers. Our machines were standard personal computers with Intel Pentium Core2 Duo CPUs, with a frequency of 2.67 GHz, and 2.00 GB of RAM. A homomorphic encryption system, the Paillier cryptosystem [12], was used to implement the protocols. The network environment in our experiments was a wireless LAN based on IEEE802.11g/IEEE802.11b.

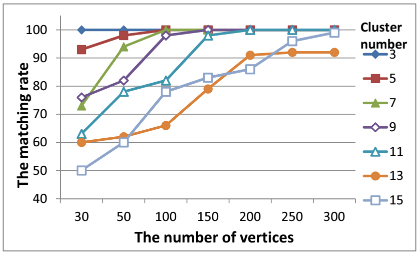


Fig. 4. Accuracy of matching

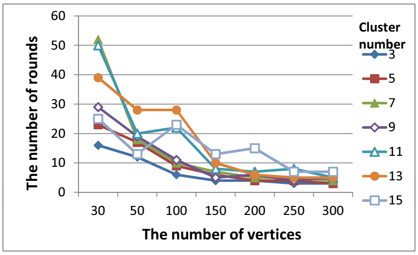


Fig. 5. Necessary number of rounds

We used artificial and real data to evaluate the accuracy and efficiency of our protocol. The artificial data were generated from generative models with different parameters. We evaluated them by comparing the inferred results with the corresponding parameters. Moreover, we selected a network of books about US politics compiled by Valdis Krebs [16] as the real data, in which nodes represented books about US politics sold by Amazon.com and edges represented the co-purchasing of books by the same buyers, as indicated by the *customers who bought this book also bought these other books* feature on Amazon. Nodes were given three labels to indicate whether they were *liberal*, *neutral*, or *conservative*. We compared our inferred results with them.

8.1 Accuracy

We executed our protocol and counted the number of results that matched the true values. We used matching rate, the percentage of matched data, to evaluate accuracy. In Fig. 4, each line expresses the relation between the number of vertices and the accuracy with respect to a special number of clusters. We found that the results could be correctly inferred with our protocol for three clusters. However, increasing the number of cluster will lead to a decrease in accuracy. Fortunately, we could increase accuracy by increasing the number of vertices. This can be verified from Fig. 4, in which each line is increasing. We also evaluated the speed of convergence by counting the number of necessary rounds of computation until convergence occurred (Fig. 5). We found that convergence became faster when there were far more vertices than numbers of clusters. We then evaluated the data set of books about US politics [16]. Although this network contains only 105 vertices and 441 pairs of edges, the accuracy was about 86%. The intuitive image of these experimental results of real data are shown in Fig. 6. We found they are quite close to the original data.

8.2 Efficiency

We used two computers in this experiment to simulate distributed computation. We executed the operators for each pair one-by-one by treating these two computers as two adjacent parties and recording the running time for each step.

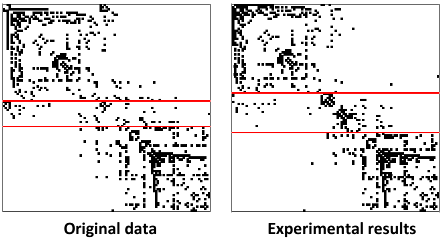


Fig. 6. Clustering result of real data

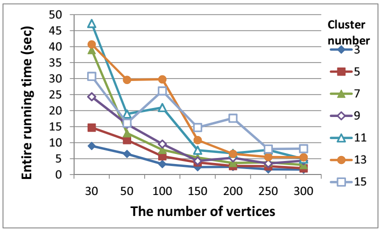


Fig. 7. Number of vertices vs. entire running time

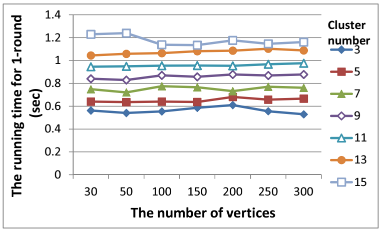


Fig. 8. Number of vertices vs. one-round of running time

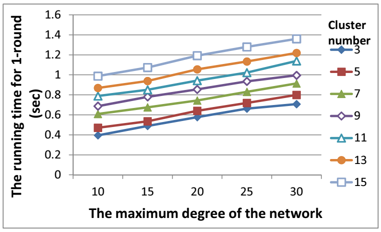


Fig. 9. Maximum degree vs. one-round of running time

We designed a parallel solution using Vizing's solution [15], and calculated the entire computational time in this parallel environment. All of our experimental results also contained the communication time. Fig. 7 plots the relation between the number of vertices and the total running time with respect to the different number of cluster. Combined with Fig. 5, we also obtained the results in Fig. 8, which illustrates one-round of running time with respect to different numbers of clusters and vertices. An interesting phenomenon is that increasing the number of vertices can decrease the entire running time (Fig. 7), although one-round running time (Fig. 8) is nearly independent of the number of vertices. This implies our privacy-preserving schema for clustering can be applied to very large-scale networks such as social networks. Fig. 9 also compares the running time with the maximum degree. The one-round of running time is increased with the increase in the maximum degree. We also found that the results in Fig. 9 agree with our description in Section 7. We also evaluated the real data using the protocol with encryption. It only needed 12 rounds of computations until convergence occurred. The entire running time was about 11 sec. That implies the average running time for one-round of computation is only about 1 sec.

9 Conclusion

We proposed a secure EM-algorithm to cluster vertices in a private network in this paper. This method deals with the mixture of assortative and disassortative mixing models. Assuming that each vertex is independent, private, and semi-honest, our algorithm was sufficiently secure to preserve the privacy of every vertex. The running time for our algorithm only depended on the number of clusters and the maximum degree. Since our algorithm does not become inefficient with larger amounts of data, it can be applied to very large-scale networks.

References

1. Newman, M.E.J., Leicht, E.A.: Grid Mixture models and exploratory analysis in networks. Proc. Natl. Acad. Sci. USA 104, 9564–9569 (2007)
2. Bunn, P., Ostrovsky, R.: Secure two-party k-means clustering. In: The 14th ACM Conference on Computer and Communications Security (2007)
3. Koller, D., Pfeffer, A.: Probabilistic frame-based systems. In: The 15th National Conference on Artificial Intelligence (1998)
4. Jha, S., Kruger, L., McDamiel, P.: Privacy preserving clustering. In: The 10th European Symposium on Research in Computer Security (2005)
5. Vaidya, J., Clifton, C.: Privacy-Preserving k-means clustering over vertically partitioned data. In: The 9th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2003)
6. Jagannathan, G., Wright, R.N.: Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In: The 11th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2003)
7. Lin, X., Clifton, C., Zhu, M.: Privacy-preserving clustering with distributed EM mixture. Knowledge and Information Systems, 68–81 (2004)
8. Hay, M., Li, C., Miklau, G., Jensen, D.: Accurate estimation of the degree distribution of private networks. In: The 9th IEEE International Conference on Data Mining (2009)
9. Liu, K., Terzi, E.: A framework for computing the privacy scores of users in online social networks. In: The 9th IEEE International Conference on Data Mining (2009)
10. Sakuma, J., Kobayashi, S.: Link analysis for private weighted graphs. In: The 32nd ACM SIGIR Conference (2009)
11. Kempe, D., McSherry, F.: A decentralized algorithm for spectral analysis. Journal of Computer and System Sciences 74(1), 70–83 (2008)
12. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
13. Kantarcoglu, M., Clifton, C.: Privacy-preserving distributed mining of association rules on horizontally partitioned data. In: The ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, DMKD 2002 (2002)
14. Goldreich, O.: Foundations of Cryptography. Basic Applications, vol. 2. Cambridge University Press, Cambridge (2004)
15. Vizing, V.G.: On an estimate of the chromatic class of a p-graph. Diskret. Analiz 3, 25–30 (1964)
16. Krebs, V.: http://www.orgnet.com/

Named Entity Recognition and Identification for Finding the Owner of a Home Page

Vassilis Plachouras[1,2], Matthieu Rivière[2], and Michalis Vazirgiannis[1,3]

[1] LIX, École Polytechnique, Palaiseau, France
vassilis.plachouras@presans.com
[2] PRESANS, X-TEC, École Polytechnique, Palaiseau, France
matthieu.riviere@presans.com
[3] Dept of Informatics, AUEB, Athens, Greece
mvazirg@aueb.gr

Abstract. Entity-based applications, such as expert search or online social networks where users search for persons, require high-quality datasets of named entity references. Obtaining such high-quality datasets can be achieved by automatically extracting metadata from Web pages. In this work, we focus on the identification of the named entity that corresponds to the owner of a particular Web page, for example, a home page or an organizational staff Web page. More specifically, from a set of named entities that have already been extracted from a Web page, we identify the one which corresponds to the owner of the home page. First, we develop a set of features which are combined in a scoring function to select the named entity of the Web page owner. Second, we formulate the problem as a classification problem in which a pair of a Web page and named entity is classified as being *associated* or not. We evaluate the proposed approaches on a set of Web pages in which we have previously identified named entities. Our experimental results show that we can identify the named entity corresponding to the owner of a home page with accuracy over 90%.

Keywords: named entity recognition, entity selection.

1 Introduction

Developing named entity-based datasets is a central task to applications such as expert search engines and scientific digital library portals, where researchers and organizations are the key entities to index and search for. However, developing such datasets is challenging because information must be extracted from unstructured or semi-structured sources. One approach involves the extraction of information from bibliographic metadata of scientific publications. DBLP[1] is an example of a site offering an index of the literature in Computer Science. CiteSeerX[2] crawls the Web to collect files that correspond to publications and

[1] http://www.informatik.uni-trier.de/~ley/db/
[2] http://citeseerx.ist.psu.edu/

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 554–565, 2012.

from which information is extracted. A complementary approach is to extract information and to identify researchers from crawled Web pages of academic institutions. By exploiting the publicly available Web sites of academic institutions, this approach has the potential to achieve higher coverage when there is no bibliographic metadata available.

In this work, we consider the latter approach to develop an entity-based dataset of researchers, covering several research fields and different countries. We describe a mechanism for identifying with high accuracy the named entity that corresponds to the owner of a Web page. In other words, given a Web page p, we identify the person entity e of the Web page's owner. Such Web pages are either home pages of people, or organizational staff Web pages, similar to online business cards. For example, the owner's named entity of a researcher's home page is the researcher's name.

Related works propose to identify the owner of a home page by learning models of the structure of home pages and the position of names on the Web page. For example, Gollapalli *et al.* [8] select the first identified name on the Web page. This simple heuristic is effective because the name of the owner of the home page is likely to appear before any other name. However, the effectiveness of this heuristic is highly dependent on the effectiveness of named entity recognition, because if the first name is not identified correctly, then the selected name is likely to be wrong.

We take a different approach, where we first apply a named entity recognizer to extract all names appearing on a Web page and then, we exploit the likely redundancy of the names' occurrences on a Web page to identify the name of the Web page's owner. For example, the name of the Web page's owner is likely to appear more than once in the Web page. In addition, it may appear both in its full form as well as in abbreviated forms in publication references. We develop weighting functions for the identified named entities and select the top-scoring ones as the named entities of a Web page's owner. The weighting functions are based on the output of the named entity recognizer and exploit similarities between names by constructing a graph whose vertices are the named entities that have been recognized in a Web page. Furthermore, we treat the problem of selecting the named entity as a binary classification problem and train an SVM classifier to identify those named entities.

An important advantage of our approach over existing ones is that it does not depend on a particular named entity recognition model. Instead, it can use any method that detects the named entities with the required granularity, that is, given names, last names and middle names. The experimental results, based on a dataset of 472 home pages manually annotated with the name of their owner, show that our proposed approaches can achieve over 87% precision in identifying the named entity of the Web page owner. When considering only the Web pages in which the correct name has been recognized at least once by the named entity recognition model, the introduced approaches achieve over 95% precision.

The remainder of this paper is organized as follows. In Section 2 we present related works from the literature. In Section 3 we briefly describe the named entity recognition method and features we employ to identify the named entities from which we select the Web page owner named entities. Section 4 introduces a framework for the selection of named entities and describes two baseline approaches and one based the construction of a graph from named entities that are similar, exploiting the redundancy in the named entity occurrences. In Section 5 we describe an approach based on supervised machine learning and, more specifically, a binary SVM classifier, which is trained to select the named entities from a home page. Section 6 describes our dataset and the experimental results we have obtained. Finally, Section 7 closes this work with some concluding remarks.

2 Related Work

The approach to identify the named entity of the home page owner is primarily related to named entity recognition, metadata extraction from Web pages, and to coreference resolution.

Named Entity Recognition. Supervised machine learning techniques are typically used to identify named entities in texts and Web pages. An example of a generative model is a Hidden Markov Model (HMM), in which the hidden states are used to model the tag classes of words. Bikel *et al.* [1] develop a named entity recognizer based on a HMM, where the hidden states of the HMM correspond to a number of name classes, such as person names, or organization names, and the features involve checking for capitalization, whether a word contains only letters, or digits. Chieu and Ng[4] employ a Maximum Entropy Classifier, which classifies each word in a text as the beginning of a named entity, the continuation, or the last word of a named entity. The features employed by the Maximum Entropy Classifier are binary. Takeuchi and Collier [14] explore the use of Support Vector Machines for named entity recognition, computing features from a context of the three previous and three following tokens. An approach that has been commonly used and results in state-of-the-art performance is Conditional Random Fields (CRF) [10], which is an undirected graphical model or a Markov Random Field. Culota *et al.* [5] extract contact information from the home pages of persons identified in email corpora. Minkow *et al.* [11] apply a CRF model to recognize names in emails, using features which are primarily based on gazetteers for person first and last names, names of organizations and locations, but not using deep natural language processing. Zhu *et al.* [17] propose two-dimensional CRF, which take into account not only the sequence of information objects in a Web page, but also the dependencies between neighboring blocks. Shi and Wang have proposed a dual-layer CRF, which aims to process more accurately cascades of subtasks in Natural Language Processing [13]. For example, one such cascade of tasks is the identification of the full person names in a text, as well as the given and last names. In our work, we employ an approach similar to cascaded CRFs where the predicted labels for the person names are used as a feature in the prediction of the first and last names of persons. We discuss in more detail the CRF model we employ in Section 3.

Metadata Extraction. While we use named entity recognition to identify person names on a Web page, the focus of our work is on selecting the name of the owner of a professional or academic home page. Hence, our work is more closely related to [9][3][8]. Kato et al. [9] employ the concept of information sender to identify the author of a Web page or the organization to which the Web page belongs. They treat the problem as a ranking problem evaluating at the top-5 results. The reported precision at ranks 1 and 5 is 0.586 and 0.752, respectively. Changuel et. al [3] extract the author of Web pages, not necessarily home pages, by building a decision tree with the C4.5 algorithm and employing a small set of features. They report a precision of approximately 0.812 in identifying the author of a web page. Gollapalli *et al.* [8] identify the owner of a home page by applying a standard named entity recognition model and selecting the first identified name. Zheng et al. [16] describe an approach based on Conditional Random Fields to identify the metadata about authors from their home pages using visual features, such as the position of DOM nodes on the rendered Web page. Finally, Tang et al. [15] start from a dataset of bibliographic metadata and create Web search engine queries to retrieve the home page of a user. Their setting is different from ours where we aim to extract the names of persons, without assuming that we have any information about the names *a priori.*

Coreference Resolution. The task of selecting the main entity from the set of entities identified on a home page is related to coreference resolution, which determines whether two textual expressions refer to the same entity or not. Typical coreference resolution methods employ supervised learning [12][6] and rely on the linguistic analysis of text to extract features. The task of identifying the owner of a home page does not require the full resolution of all references, and hence, it is not necessary to apply coreference resolution at a first step.

3 Named Entity Recognition

Before presenting our approach to named entity selection, we describe the Named Entity Recognition (NER) system we first apply to extract names from home pages. We have developed a NER system based on supervised learning of a Conditional Random Field (CRF) to learn to recognize the full names of persons, as well as their first, middle and last names. We did not employ an existing NER system such as the Stanford Named Entity Recognizer[3] [7] for two main reasons. First, we require better granularity in identifying first, last and middle names in addition to full names. Second, our objective is to process input from Web pages. Hence, we develop features that exploit term frequency statistics in the anchor text of incoming hyperlinks of Web pages.

To train the CRF model, we have manually annotated all the names in 95 Web pages. We first split the textual content of a Web page in sentences using the DOM tree and regular expressions. Next, we tokenize each sentence by splitting tokens at non-alphanumeric characters, and we annotate the tokens. We use

[3] Available from `http://nlp.stanford.edu/software/CRF-NER.shtml`

the *begin, inside, outside* (BIO) convention for labels. For example the sentence *"Chris Bishop is a Distinguished Scientist at ..."*[4] is tokenized and labeled as follows:

Chris	Bishop	is	a	Distinguished	Scientist	at ...
`BPERSON`	`IPERSON`	`O`	`O`	`O`	`O`	`O`
`BFNAME`	`BLNAME`	`O`	`O`	`O`	`O`	`O`

where `BPERSON` denotes the beginning of a full name, `BFNAME` denotes the beginning of a first name, and `BLNAME` denotes the beginning of a last name. The label `IPERSON` denotes that the corresponding token is inside a person name. The label `O` denotes that the corresponding token does not belong to any of the classes we consider.

Next, we train a CRF model using five types of features. The first type of features corresponds to the tokens themselves. The second type corresponds to two features, whose value depends on the form of the examined token. The first feature indicates whether the token contains only numerical digits, or it is a single upper case letter, or a punctuation symbol, or a capitalized word, *etc.*). The second feature indicates whether the token is an alphanumeric string. The third type of features relies on two gazetteers for first names and geographic locations, respectively, and comprises two binary features indicating whether the token is a first name, and whether the token is a geographic location. The fourth type of features is based on a full-text index of Web pages and comprises 4 features. More specifically, two features correspond to the logarithm of the number of documents in which the term occurs in the body, and the anchor text of incoming links respectively. The two next features correspond to flags indicating whether the term occurs in the title or the anchor text of incoming links of the currently processed document. The fifth type of features comprises one feature indicating whether the token occurs in the anchor text of an outgoing hyperlink in the currently processed document, differentiating between links to Web pages in the same or different domains. Note that the last two types of features depend on the distribution of terms in a full text index of Web pages, and the text associated with the link structure of Web pages. We employ the implementation of `CRF++`[5].

We learn the CRF model and apply it to unseen Web pages in the following way. First, we train a CRF to recognize full names and assign labels `BPERSON` and `IPERSON` . The assigned labels are then used to learn a second model where the assigned labels constitute an additional twelfth feature used in the recognition of first, middle and last names, assigning labels `BFNAME`, `IFNAME`, `BMNAME`, `IMNAME`, and `BLNAME` and `ILNAME`, respectively. After applying the CRF models to label tokens, we aggregate consecutive tokens with B and I labels in an entity e. For each entity e, $t(e)$ is the type of the entity where $t(e) \in \{\texttt{PERSON}, \texttt{FNAME}, \texttt{MNAME}, \texttt{LNAME}\}$, $c(e)$ is the average confidence of the label assignment over the entity's tokens, and $s(e)$ is the concatenation of the tokens to form the string representation of e.

[4] Quoted from `http://research.microsoft.com/en-us/um/people/cmbishop/`

[5] Available from `http://crfpp.sourceforge.net/`

The accuracy of the named entity recognition could potentially be higher if we employed language-specific features, such as Part Of Speech (POS) tags. However, our aim is to apply the developed approaches to a wide range of input Web pages, irrespectively of the language they are written in. We offset the potentially lower accuracy of the CRF named entity recognition by weighting the different occurrences of names, as described in the following section.

4 Finding Named Entities of Web Page Owner

In this section, we study the problem of selecting the named entity corresponding to the owner of a home page. We operate on the output of the named entity recognition process described in Section 3 to select the entity e with type $t(e) = PERSON$. First, we describe the framework for weighting the identified entities (Section 4.1). Then, we introduce two baseline weighting functions for entities based on the features used by the NER system (Section 4.2), and a third weighting function based on a graph representation of the named entities (Section 4.3).

4.1 Entity Selection Framework

We perform entity selection in the following framework. $S(t, str)$ is the set of all entities of type $t(e) = t$ and string representation $s(e) = str$:

$$S(str) = \{e|t(e) = t \wedge s(e) = str\} \tag{1}$$

When $t = \texttt{PERSON}$ we write $S(str) = S(\texttt{PERSON}, str)$. For each set $S(str)$, we compute a weight w_{str} and rank $S(str)$ in descending order of w_{str}. The selected named entities of the processed Web page are the ones belonging to the top ranked $S(str)$.

4.2 Baseline Entity Selection

A simple way to weight a set $S(str)$ of PERSON entities with the same string representation is to sum the confidence $c(e)$ of the label assignment for each entity $e \in S(str)$:

$$w_{str} = \sum_{e \in S(str)} c(e) \tag{2}$$

The intuition for defining the weight w_{str} as the sum of the confidences is that it reflects both the number of times the same string has been identified as a PERSON entity as well as the confidence in the recognition.

The weighting of $S(str)$ from Eq. 2 is only based on the average confidence of the label assignment to each token of the entities in $S(str)$. We can improve the weighting by incorporating more information regarding the position of the occurrences of entities.

$$w_{str} = \sum_{e \in S(str)} (w_a anchor(e) + w_t title(e) + w_c c(e)) \tag{3}$$

where $anchor(e) = 1$ if $s(e)$ occurs in the anchor text of incoming hyperlinks of the processed Web page, otherwise $anchor(e) = 0$. Similarly, $title(e) = 1$ if $s(e)$ occurs in the title of the processed Web page, otherwise $title(e) = 0$. The parameters w_a, w_t, w_c control the importance of each of the three features and are set during training.

4.3 Graph-Based Entity Selection

The baseline weighting of set $S(str)$ according to Eq. 2 and 3 only consider the entities of type PERSON with the same string representation. However, they ignore any similarities between the identified entities in order to compute an improved weight. Suppose that on a Web page the full name of a researcher appears only twice at the top of the Web page, and the name of the most frequent co-author appears in abbreviated form once for each publication of the researcher[6]. In such a setting, the baseline weighting functions may select the abbreviated name of the co-author as the named entity of the URL's owner, instead of the full name of the researcher.

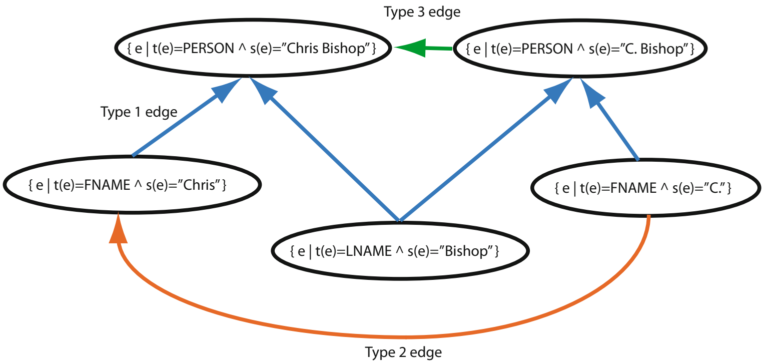

Type 3 edge
{ e | t(e)=PERSON ^ s(e)="Chris Bishop"}
{ e | t(e)=PERSON ^ s(e)="C. Bishop"}
Type 1 edge
{ e | t(e)=FNAME ^ s(e)="Chris"}
{ e | t(e)=FNAME ^ s(e)="C."}
{ e | t(e)=LNAME ^ s(e)="Bishop"}
Type 2 edge

Fig. 1. The graph constructed from the sets of identified named entities in a Web page

We overcome the limitations of the baseline weightings by introducing a novel graph-based weighting for sets of entities. We define a directed graph $G = \{V, E\}$ where V is the set of vertexes and E is the set of edges. Each set $S(t, str) = \{e|t(e) = t \wedge s(e) = str\}$ of entities with given type t and string representation str, corresponds to a vertex of V. Hence, the graph is constructed from all identified names in the Web page.

We define three types of directed edges in graph G. The set of vertices having an edge of type i to $S(t, str)$ is denoted by $in_i(t, str)$. When $t =$ PERSON, we can write $in_i(str)$. The three types of directed edges are defined as follows:

[6] For example, http://www.cs.washington.edu/homes/pedrod/

- A type 1 edge connects sets of `FNAME, MNAME, LNAME` entities to the corresponding sets of `PERSON` entities in which they occur.
- A type 2 edge connects a set $S(t, str1)$ to $S(t, str2)$ when string $str1$ is an abbreviated form of $str2$ and $t \in \{\texttt{FNAME, MNAME, LNAME}\}$.
- A type 3 edge connects a set $S(str1)$ to $S(str2)$ when the name $str1$ is an abbreviated form of the name $str2$. Formally, $S(str1) \in in_3(S(str2))$ if there exists $S(t, str3) \in in_1(\texttt{PERSON}, str1) \cap in_1(\texttt{PERSON}, str2)$ and there exist $S(t', str4) \in in_1(\texttt{PERSON}, str1))$, $S(t'', str5) \in in_1(\texttt{PERSON}, str2)$ where $S(t', str4) \in in_2(t'', str5)$.

Figure 1 illustrates a graph constructed from a set of identified named entities. The graph has two vertexes of type `PERSON`, one vertex of type `LNAME` for the last name *'Bishop'* and two vertexes of type `FNAME` for the first name *'Chris'* and its abbreviated form *'C.'* There are four edges of type 1, linking the vertexes of type `FNAME` and `LNAME` to the corresponding vertexes of type `PERSON`. There is one edge of type 2 which links the vertex $S(\texttt{FNAME}, \text{'C.'})$ to the vertex $S(\texttt{FNAME}, \text{'Chris'})$. Finally, there is one edge of type 3 from $S(\text{'C. Bishop'})$ to $S(\text{'Chris Bishop'})$ because both vertexes have incoming links from the same vertex $S(\texttt{LNAME}, \text{'Bishop'})$ and there is a type 2 edge between two of their `FNAME` linking vertexes.

The graph G, which is constructed as described above, is a directed acyclic graph (DAG). From the definition of type 1 edges, we cannot have a cycle involving vertices of type `PERSON` and any other entity type because type 1 edges always point to vertices of type `PERSON`. Hence, a cycle may involve either type 2 edges exclusively or type 3 edges exclusively. Since a type 3 edge exists only if there is a type 2 edge, and the two edges cannot be in the same path, then there exists a cycle with type 3 edges only if there exists a cycle with type 2 edges. However, there cannot be a cycle with type 2 edges, because type 2 edges link an abbreviated name to its full form. Hence, there cannot be any cycle in the graph G.

Once we have constructed the graph from the named entities identified in a Web page, we compute a weight for each vertex $S(str)$, corresponding to the sum of the Baseline 2 score from Eq. 3 plus the sum of the scores of vertices that link to $S(str)$.

$$w_{str} = \sum_{e \in S(str)} (w_a anchor(e) + w_t title(e) + w_c c(e)) + \sum_{S(str') \in in_i(str)} w_{str'} \quad (4)$$

Finally, we select the set $S(str)$ with the highest score w_{str} according to Eq. 4. The intuition is that the scores of abbreviated named entities propagate to the entities corresponding to full names.

5 Learning to Select Named Entities

The baseline and the graph-based scoring functions make use of the output of the NER system to score entities found in a Web page and select the ones which are more likely to refer to the owner of the Web page. However, all three functions will always produce a score for the entities, even when the named entity of the owner of the Web page is not among the identified named entities. For example, a researcher may have a set of Web pages documenting a software he has written and released as open-source. The functions introduced earlier will always select one set of entities as the owner for the considered Web page. Moreover, extending these functions with arbitrary features is not trivial. In this section, we investigate the problem of selecting the named entities as a binary classification problem in a supervised learning setting.

In particular, we formulate the classification problem $y(x) \in \{-1, 1\}$, where $x \in \mathcal{X} = \{(\text{URL}, S(\texttt{PERSON}, str))\}$. The input x is a pair of a URL and a set $S(\texttt{PERSON}, str)$. For the output, $y(x) = 1$ when the named entities in $S(\texttt{PERSON}, str)$ correspond to the owner of Web page with URL, otherwise, $y(x) = -1$. For each input point x, we compute 13 features:

- the graph-based score of $S(\texttt{PERSON}, str)$ from Eq. 4
- the rank of $S(\texttt{PERSON}, str)$ when all sets of PERSON entities are ordered in ascending order of the Baseline 1, Baseline 2, graph-based scoring functions, as well as in the order of occurrence (4 features)
- the sum of the cardinalities $|S(t, str')|$ where $S(t, str') \in in_i(\texttt{PERSON}, str)$ for each type of links (3 features)
- the number of edges of type i pointing to $S(\texttt{PERSON}, str)$ for $i = 1, 2, 3$ (3 features)
- 1 if *str* appears in an email address found in the content of URL, otherwise 0
- 1 if *str* appears to be emphasized in the text of home page identified by URL, otherwise 0

The feature values are normalized between -1 and +1 on a per home page basis. We employ an SVM classifier with radial-basis kernel from LIBSVM[7] [2]. For a given home page identified by URL, if the SVM classifies as +1 more than one pairs $(\text{URL}, S(\texttt{PERSON}, str))$, we select the one with the highest estimated probability, as computed by the SVM classifier.

6 Experimental Results

In this section, we describe the experimental setting in which we evaluate the introduced methods. First, we evaluate the CRF-based named entity recognition (Section 6.1). Next, we describe the dataset we use for entity selection and we present the obtained results (Section 6.2).

6.1 Named Entity Recognition Evaluation

In this section, we present evaluation results for the NER system we describe in Section 3. Starting from a set of 95 annotated Web pages, we randomize their

[7] Available from `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

order and split them in three folds. We use each fold once to test the CRF model we learn on the other two folds. Table 1 reports the micro-averaged precision, recall and F-measure for each of the labels we assign during the first and the second passes of the CRF-based NER system, respectively.

The NER system assigns BPERSON and IPERSON labels with high precision and recall. This is consistent with results reported for NER systems trained on much larger corpora [7]. First and last names are identified with an accuracy of more than 0.80. The obtained precision for middle names is significantly lower, mainly due to the small number of training examples available.

Table 1. Number of annotated tokens, micro-averaged precision, recall, and F-measure for each of the labels assigned in the first and second passes, respectively

Label	# of Annotated Tokens	Precision	Recall	F-Measure
	Pass 1			
BPERSON	3326	0.931	0.918	0.924
IPERSON	6552	0.955	0.913	0.934
O	35364	0.978	0.988	0.983
	Pass 2			
BFIRST	2942	0.820	0.900	0.858
BLAST	2956	0.818	0.879	0.848
BMIDDLE	131	0.290	0.344	0.315
IFIRST	1783	0.820	0.871	0.844
ILAST	777	0.832	0.793	0.812
IMIDDLE	120	0.542	0.375	0.443
O	36533	0.975	0.960	0.967

6.2 Evaluation and Experimental Results

We have evaluated the introduced approaches using a dataset of home pages, for which we have manually identified the full name, as well as the first, middle and last names of the home page owners. We have sampled a total of 472 home pages from a large crawl of university and research organization Web sites.

The NER model, described in Section 3, has identified the correct name at least once in 432 out of the 472 home pages. Out of the 432 pages, 66% of the pages are written in English, 27% are written in French and 3% of the pages are written in German. The remaining 4% of the pages are written in Danish, Italian, Polish, Portuguese and Swedish. We distinguish between perfect and partial matches of names. We have a perfect match when the entity weighting ranks first a name matching perfectly the correct one. A partial match occurs when the entity weighting ranks first an abbreviated version of the correct name.

Table 2 reports the accuracy of perfect and partial identifications over the 432 home pages for which the correct answer is among the identified named entities. We also report results computed over the total number of home pages. The first approach (Order) is a naïve heuristic where the first identified person name is selected as the owner's name for the corresponding page. The effectiveness of this heuristic depends on the accuracy of the underlying NER system because any wrong identification of names will lead to an error in the selection [8]. The two next approaches, Baseline 1 and Baseline 2, correspond to the selection of entities using Eq. 2 and 3, respectively. The fourth and fifth rows in Table 2 display the results obtained with the graph-based and the SVM-based entity selection approaches, respectively.

Table 2. Fraction of Web pages for which there is a perfect or partial match, when using Order, Baseline 1, Baseline 2, Graph and SVM-based entity selection

	Perfect	Partial	Perfect+Partial	(Perfect+Partial)/All Pages
Order	0.847	0.035	0.882	0.807
Baseline 1	0.789	0.090	0.880	0.805
Baseline 2	0.875	0.039	0.914	0.837
Graph-based	0.944	0.014	0.958	0.877
SVM-based	0.954	0.009	0.963	0.881

The best-performing approach is the SVM-based one, which achieves perfect matches in 95.4% of the home pages when the named entity recognition identifies the correct name at least once. If we consider both perfect and partial matches, then we have a match in 96.3% of the home pages. When we calculate the results on all the home pages, including the ones in which named entity recognition did not identify the correct named entity, we achieve a precision of 88.1%.

7 Conclusions

In this work, we have introduced a novel method to select among recognized named entities in a home page the one corresponds to the owner. Our method uses the output of a named entity recognition system and exploits the redundancy and the similarities between names to select the correct one. The introduced methods are developed independently of the employed named entity recognition approach. Indeed, they can be used with any NER approach that identifies person names, but also first, middle and last names. In a dataset of more than 400 home pages, our methods identify the correct name for more than 90% of the home pages in which a NER system identifies at least once the correct name in the processed page. The comparison of our methods with a heuristic based on the order of names shows that our approaches achieve important improvements in effectiveness because they are more robust with respect to the accuracy of the employed NER system.

We have applied the developed methods in the context of researchers' home pages. In the future, we will evaluate it in the context of different applications, such as the automatic creation of online social networks, or people search. We also aim to apply the developed methods for identifying the name of the owner of a Web page as a feature to improve the classification of Web pages.

References

1. Bikel, D.M., Miller, S., Schwartz, R., Weischedel, R.: Nymble: a high-performance learning name-finder. In: Procs. of the 5th ANLC, pp. 194–201 (1997)
2. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. ACM Trans. Intell. Syst. Technol. 2, 27:1–27:27 (2011)
3. Changuel, S., Labroche, N., Bouchon-Meunier, B.: Automatic web pages author extraction. In: Procs. of the 8th FQAS, pp. 300–311 (2009)
4. Chieu, H.L., Ng, H.T.: Named entity recognition with a maximum entropy approach. In: Procs. of the 7th Conference on Natural Language Learning at HLT-NAACL 2003, CONLL 2003, vol. 4, pp. 160–163 (2003)
5. Culotta, A., Bekkerman, R., McCallum, A.: Extracting social networks and contact information from email and the web. In: CEAS (2004)
6. Culotta, A., Wick, M., Hall, R., McCallum, A.: First-order probabilistic models for coreference resolution. In: Procs. of HLT/NAACL, pp. 81–88 (2007)
7. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Procs. of the 43rd Annual Meeting on ACL, pp. 363–370 (2005)
8. Gollapalli, S.D., Giles, C.L., Mitra, P., Caragea, C.: On identifying academic homepages for digital libraries. In: Procs. of the 11th JCDL, pp. 123–132 (2011)
9. Kato, Y., Kawahara, D., Inui, K., Kurohashi, S., Shibata, T.: Extracting the author of web pages. In: Procs. of the 2nd ACM WICOW, pp. 35–42 (2008)
10. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Procs. of the 18th ICML, pp. 282–289 (2001)
11. Minkov, E., Wang, R.C., Cohen, W.W.: Extracting personal names from email: applying named entity recognition to informal text. In: Procs. of the Conf. on HLT and EMNLP, HLT 2005, pp. 443–450 (2005)
12. Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In: Procs. of the 40th Annual Meeting on ACL, ACL 2002, pp. 104–111 (2002)
13. Shi, Y., Wang, M.: A dual-layer crfs based joint decoding method for cascaded segmentation and labeling tasks. In: Procs. of the 20th IJCAI, pp. 1707–1712 (2007)
14. Takeuchi, K., Collier, N.: Use of support vector machines in extended named entity recognition. In: Procs. of the 6th Conference on Natural Language Learning, COLING 2002, vol. 20, pp. 1–7 (2002)
15. Tang, J., Zhang, D., Yao, L.: Social network extraction of academic researchers. In: Procs. of the 7th ICDM, pp. 292–301 (2007)
16. Zheng, S., Zhou, D., Li, J., Giles, C.L.: Extracting author meta-data from web using visual features. In: Procs. of the 7th ICDMW, pp. 33–40 (2007)
17. Zhu, J., Nie, Z., Wen, J.R., Zhang, B., Ma, W.Y.: 2d conditional random fields for web information extraction. In: Procs. of the 22nd ICML, pp. 1044–1051 (2005)

Clustering and Understanding Documents via Discrimination Information Maximization

Malik Tahir Hassan and Asim Karim

Dept. of Computer Science, LUMS School of Science and Engineering
Lahore, Pakistan
{mhassan,akarim}@lums.edu.pk

Abstract. Text document clustering is a popular task for understanding and summarizing large document collections. Besides the need for efficiency, document clustering methods should produce clusters that are readily understandable as collections of documents relating to particular contexts or topics. Existing clustering methods often ignore term-document semantics while relying upon geometric similarity measures. In this paper, we present an efficient iterative partitional clustering method, CDIM, that maximizes the sum of discrimination information provided by documents. The discrimination information of a document is computed from the discrimination information provided by the terms in it, and term discrimination information is estimated from the currently labeled document collection. A key advantage of CDIM is that its clusters are describable by their highly discriminating terms – terms with high semantic relatedness to their clusters' contexts. We evaluate CDIM both qualitatively and quantitatively on ten text data sets. In clustering quality evaluation, we find that CDIM produces high-quality clusters superior to those generated by the best methods. We also demonstrate the understandability provided by CDIM, suggesting its suitability for practical document clustering.

1 Introduction

Text document clustering discovers groups of related documents in large document collections. It achieves this by optimizing an objective function defined over the entire data collection. The importance of document clustering has grown significantly over the years as the world moves toward a paperless environment and the Web continues to dominate our lives. Efficient and effective document clustering methods can help in better document organization (e.g. digital libraries, corporate documents, etc) as well as quicker and improved information retrieval (e.g. online search).

Besides the need for efficiency, document clustering methods should be able to handle the large term space of document collections to produce readily understandable clusters. These requirements are often not satisfied in popular clustering methods. For example, in K-means clustering, documents are compared in the term space, which is typically sparse, using generic similarity measures without considering the term-document semantics other than their vectorial representation in space. Moreover, it is not straightforward to interpret and understand the clusters formed by K-means clustering; the similarity of a document to its cluster's mean provides little understanding of the document's context or topic.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 566–577, 2012.

In this paper, we present a new document clustering method based on discrimination information maximization (CDIM). CDIM's semantically motivated objective function is maximized via an efficient iterative procedure that repeatedly projects documents onto a K-dimensional discrimination information space and assigns documents to the cluster along whose axis they have the largest value. The discrimination information space is defined by term discrimination information estimated from the labeled document collection produced in the previous iteration. This procedure maximizes the sum of discrimination information provided by all documents. A key advantage of using term discrimination information is that each cluster can be interpreted by a list of highly discriminating terms. These terms serve as units of understanding, as demonstrated in linguistics studies [1,2], describing a cluster in the document collection. We evaluate the performance of CDIM on ten popular text data sets. In clustering quality evaluation, CDIM is found to produce high quality clusters superior to those produced by non-negative matrix factorization (NMF) and several K-means variants. Our results suggest the practical suitability of CDIM for clustering and understanding of document collections.

The rest of the paper is organized as follows. We discuss the related work and motivation for our method in Section 2. CDIM, our document clustering method is described in detail in Section 3. Section 4 presents our experimental setup. Section 5 discusses the results of our experiments, and we conclude with future directions in Section 6.

2 Motivation and Related Work

Content-based document clustering continues to be challenging because of (1) the high dimensionality of the term-document space, (2) the sparsity of the documents in the term-document space, and (3) the difficulty of incorporating appropriate term-document semantics for improved clustering quality and understandability. Moreover, real-world document clustering often involves large document collections thus requiring the clustering method to be efficient.

The K-means algorithm continues to be popular for document clustering due to its efficiency and ease of implementation [3]. It is a partitional clustering method that optimizes an objective function via an iterative two-step procedure. Usually, documents are represented by terms' weights, and documents are compared in the term space by the cosine similarity measure. Several clustering objective functions can be optimized [4] with the traditional objective of maximizing the similarity of documents to their cluster means producing reliable clusterings. The Repeated Bisection clustering method, which splits clusters into two until the desired number of clusters are obtained, has been shown to produce better clusterings especially when K is large (greater than 20) [5]. These K-means based methods are efficient and accurate for many practical applications. Their primary shortcoming is poor interpretability of the clusters where the cluster mean vector is often not a reliable indicator of the documents in a cluster.

Some researchers have used external knowledge bases to semantically enrich the document representation for document clustering [6,7]. In [6], Wikipedia's concepts and categories are adopted to enhance the document representation, while in [7] several ontology-based (e.g. WordNet) term relatedness measures are evaluated for semantically smoothing the document representation. In both works, it has been shown that the quality of clusterings produced by the K-means algorithm improves over the baseline ("bag of words") document representation. However, extracting information from knowledge bases is computationally expensive. Furthermore, these approaches suffer from the same shortcomings of K-means regarding cluster understandability.

The challenge of high dimensional data clustering, including that of document clustering, has been tackled by clustering in a lower dimensional space of the original term space. One way to achieve this is through Non-Negative Matrix Factorization (NMF). NMF approximates the term-document matrix by the product of term-cluster and document-cluster matrices [8]. Extensions to this idea, with the goal of improving the interpretability of the extracted clusters, have also been proposed [9,10]. Another way is to combine clustering with dimensionality reduction techniques [11,12]. Nonetheless, these methods are restricted by their focus on approximation rather than semantically useful clusters, and furthermore, dimensionality reduction based techniques are often computationally expensive.

Recently, it has been demonstrated that the relatedness of a term to a context or topic in a document collection can be quantified by its discrimination information [2]. Such a notion of relatedness, as opposed to the traditional term-to-term relatedness, can be effectively used for data mining tasks like classification [13]. Meanwhile, measures of discrimination information, such as relative risk, odds ratio, risk difference, and Kullback-Leibler divergence, are gaining popularity in data mining [14,15]. In the biomedical domain, on the other hand, measures like relative risk have been used for a long time for cohort studies and factor analysis [16,17].

3 CDIM – Our Document Clustering Method

CDIM (Clustering via Discrimination Information Maximization) is an iterative partitional document clustering method that finds K groups of documents in a K-dimensional discrimination information space. It does this by following an efficient two-step procedure of document projection and assignment with the goal of maximizing the sum of documents' discrimination scores. CDIM's clusters are describable by highly discriminating terms related to the context/topic of the documents in the cluster. We start our presentation of CDIM by formally stating the problem.

3.1 Problem Statement

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \Re^{M \times N}$ be the term-document matrix in which the ith document $\mathbf{x}_i = [x_{1i}, x_{2i}, \ldots, x_{Mi}]^T$ is represented by an M-dimensional vector (ith column of matrix $\mathbf{X}$). M is the total number of distinct terms in the N documents. The weight of term j in document i, denoted by x_{ji}, is equal to the count of term j in document i.

Our goal is to find K (usually in practice $K \ll \min\{M, N\}$) clusters $\mathcal{C}_k$ ($k = 1, 2, \ldots, K$) of documents such that if a document $\mathbf{x} \in \mathcal{C}_k$ then $\mathbf{x} \notin \mathcal{C}_j, \forall j \neq k$. Thus, we assume hard partitioning of the documents among the clusters; however, this assumption can be relaxed trivially in CDIM but we do not discuss this further in our current work. In addition to the cluster composition, we will also like to find significant describing terms for each cluster. Let $\mathcal{T}_k$ be the index set of significant terms for cluster k.

3.2 Clustering Objective Function

CDIM finds K clusters in the document collection by maximizing the sum of discrimination scores of documents for their respective clusters. If we denote the discrimination information provided by document i for cluster k by d_{ik} and the discrimination information provided by document i for all clusters but cluster k by $\bar{d}_{ik}$, then the discrimination score of document i for cluster k is defined as $\hat{d}_{ik} = d_{ik} - \bar{d}_{ik}$. CDIM's objective function can then be written as

$$J = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathcal{C}_k} r_{ik}(d_{ik} - \bar{d}_{ik}) \tag{1}$$

where $r_{ik} = 1$ if document i is assigned to cluster k and zero otherwise. Document discrimination information (d_{ik} and $\bar{d}_{ik}$) is computed from term discrimination information that in turn is estimated from the current labeled document collection. These computations are discussed in the following subsections.

Intuitively, CDIM seeks a clustering in which the discrimination information provided by documents for their cluster is higher than the discrimination information provided by them for the remaining clusters. It is not sufficient to maximize just the discrimination information of documents for their respective clusters as they may also provide high discrimination information for the remaining clusters.

The objective function J is maximized by using a greedy two-step procedure. In one step, given a cluster assignment defined by $r_{ik}, \forall i, k$, J is maximized by estimating $d_{ik}, \forall i, k$ and $\bar{d}_{ik}, \forall i, k$ from the labeled document collection. This estimation is done using maximum likelihood estimation. In the other step, given estimated discrimination scores $\hat{d}_{ik}, \forall i, k$ of documents, J is maximized by assigning each document to the cluster k for which the document's discrimination score is maximum. This two-step procedure continues until the change in J from one iteration to the next drops below a specified threshold value. Convergence is guaranteed because J is non-decreasing from one iteration to the next and J is upper-bounded by a local maxima.

3.3 Term Discrimination Information

The discrimination information provided by a document is computed from the discrimination information provided by the terms in the document. The discrimination information provided by a term for cluster k is quantified with the relative risk of the term for cluster k over the remaining clusters. Mathematically, the discrimination information of term j for cluster k and term j for all clusters but k is given by

$$w_{jk} = \begin{cases} \frac{p(x_j|\mathcal{C}_k)}{p(x_j|\bar{\mathcal{C}}_k)} & \text{when } p(x_j|\mathcal{C}_k) - p(x_j|\bar{\mathcal{C}}_k) > t \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \tag{2}$$

$$\bar{w}_{jk} = \begin{cases} \frac{p(x_j|\bar{\mathcal{C}}_k)}{p(x_j|\mathcal{C}_k)} & \text{when } p(x_j|\bar{\mathcal{C}}_k) - p(x_j|\mathcal{C}_k) > t \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $p(x_j|\mathcal{C}_k)$ is the conditional probability of term j in cluster k and $\bar{\mathcal{C}}_k$ denotes all clusters but cluster k. The term discrimination information is either zero (no discrimination information) or greater than one with a larger value signifying higher discriminative power. The conditional probabilities in Equations 2 and 3 are estimated via smoothed maximum likelihood estimation.

3.4 Relatedness of Terms to Clusters

In Equations 2 and 3, $t \geq 0$ is a term selection parameter that controls the exclusion of terms that provide insignificant discrimination information. As the value of t is increased from zero, fewer terms will have a discrimination information greater than one.

The index set of terms that provide significant discrimination information for cluster k ($\mathcal{T}_k$) is defined as $\mathcal{T}_k = \{j|w_{jk} > 0, \forall j\}$. These terms and their discrimination information provide a good understanding of the context of documents in cluster k in contrast with those in other clusters in the document collection. In general, $\mathcal{T}_k \cap \mathcal{T}_j \neq \emptyset, \forall j \neq k$. That is, there may be terms that provide significant discrimination information for more than one cluster. Also, depending on the value of t, there may be terms that do not provide significant discrimination information for all clusters.

In a study discussed in [1], it has been shown that humans comprehend text by associating terms with particular contexts or topics. These relationships are different from the traditional lexical relationships (e.g synonymy, antonymy, etc), but are more fundamental in conveying meaning and understanding. Recently, it has been shown that the degree of relatedness of a term to a context is proportional to the term's discrimination information for that context in a corpus [2]. Given these studies, we can consider all terms in $\mathcal{T}_k$ to be related to cluster k and the strength of this relatedness is given by the term's discrimination information. This is an important characteristic of CDIM whereby each cluster's context is describable by a set of related terms. Furthermore, these terms and their weights (discrimination information) define a K-dimensional space in which documents are comparable by their discrimination information.

3.5 Document Discrimination Information

A document i is describable by the terms it contains. Each term j in the document vouches for the context or cluster k according to the value of the term's discrimination information w_{jk}. Equivalently, each term j in the document has a certain degree of relatedness to context or cluster k according to the value w_{jk}. The discrimination information provided by document i for cluster k can be computed as the average term discrimination information for cluster k:

$$d_{ik} = \frac{\sum_{j \in \mathcal{T}_k} x_{ji} w_{jk}}{\sum_{j \in \mathcal{T}_k} x_{ji}}. \tag{4}$$

A similar expression can be used to define $\bar{d}_{ik}$. The document discrimination information d_{ik} can be thought of as the relatedness (discrimination) of document i to cluster k. The document discrimination score is given by $\hat{d}_{ik} = d_{ik} - \bar{d}_{ik}$; the larger this value is, the more likely that document i belongs to cluster k. Note that a term contributes to the discrimination information of document i for cluster k only if it belongs to $\mathcal{T}_k$ and it occurs in document i. If such a term occurs multiple times in the document then each of its occurrence contributes to the discrimination information. Thus, the discrimination information of a document for a particular cluster increases with the increase in occurrences of highly discriminating terms for that cluster.

3.6 Algorithm

CDIM can be described more compactly in matrix notation. CDIM's algorithm, which is outlined in Algorithm 1, is described next.

Let $\mathbf{W}$ ($\bar{\mathbf{W}}$) be the $M \times K$ matrix formed from the elements $w_{jk}, \forall j, k$ ($\bar{w}_{jk}, \forall j, k$), $\hat{\mathbf{D}}$ be the $N \times K$ matrix formed from the elements $\hat{d}_{ik}, \forall i, k$, and $\mathbf{R}$ be the $N \times K$ matrix formed from the elements $r_{ik}, \forall i, k$. At the start, each document is assigned to one of the K randomly selected seeds using cosine similarity, thus defining the matrix $\mathbf{R}$. Then, a loop is executed consisting of two steps. In the first step, the term discrimination information matrices ($\mathbf{W}$ and $\bar{\mathbf{W}}$) are estimated from the term-document matrix $\mathbf{X}$ and the current document assignment matrix $\mathbf{R}$. The second step projects the documents onto the relatedness or discrimination score space to create the discrimination score matrix $\hat{\mathbf{D}}$. Mathematically, this transformation is given by

$$\hat{\mathbf{D}} = (\mathbf{X}\mathbf{\Sigma})^T(\mathbf{W} - \bar{\mathbf{W}}) \tag{5}$$

where $\mathbf{\Sigma}$ is a $N \times N$ diagonal matrix defined by elements $\sigma_{ii} = 1/\sum_j x_{ji}$. The matrix $\hat{\mathbf{D}}$ represents the documents in the K-dimensional discrimination score space.

Documents are re-assigned to clusters based on their discrimination scores. A document i is assigned to cluster k if $\hat{d}_{ik} \geq \hat{d}_{ij}, \forall j \neq k$ (ties are broken arbitrarily). In matrix notation, this operation can be written as

$$\mathbf{R} = \text{maxrow}(\hat{\mathbf{D}}) \tag{6}$$

where 'maxrow' is an operator that works on each row of $\hat{\mathbf{D}}$ and returns a 1 for the maximum value and a zero for all other values. The processing of Equations 5 and 6 are repeated until the absolute difference in the objective function becomes less than a specified small value. The objective function J is computed by summing the maximum values from each row of matrix $\hat{\mathbf{D}}$.

The algorithm outputs the final document assignment matrix $\mathbf{R}$ and the final term discrimination information matrix $\mathbf{W}$. It is easy to see that the computational time complexity of CDIM is $O(KMNI)$ where I is the number of iterations required to reach the final clustering. Thus, the computational time of CDIM depends linearly on the clustering parameters.

4 Experimental Setup

Our evaluations comprise of two sets of experiments. First, we evaluate the clustering quality of CDIM and compare it with other clustering methods on 10 text data sets. Second, we illustrate the understanding that is provided by CDIM clustering. The results of these experiments are given in the next section. Here, we describe our experimental setup.

Algorithm 1. CDIM – Document Clustering via Discrimination Information Maximization

```
Require: $\mathbf{X}$ (term-document matrix), $K$ (no. of clusters)
$\mathbf{R}^{(0)} \leftarrow$ initial assignment of documents to clusters
$\tau \leftarrow 0$
$J^{(0)} \leftarrow 0$
repeat
    $\mathbf{W}^{(\tau)}, \bar{\mathbf{W}}^{(\tau)} \leftarrow$ term discrimination info estimated from $\mathbf{X}$ and $\mathbf{R}^{(\tau)}$ (Eqs. 2 and 3)
    $\hat{\mathbf{D}}^{(\tau+1)} \leftarrow (\mathbf{X}\mathbf{\Sigma})^T(\mathbf{W}^{(\tau)} - \bar{\mathbf{W}}^{(\tau)})$
    $\mathbf{R}^{(\tau+1)} \leftarrow \mathrm{maxrow}(\hat{\mathbf{D}}^{(\tau+1)})$
    $J^{(\tau+1)} \leftarrow$ sum of max discrimination scores from each row of $\hat{\mathbf{D}}^{(\tau+1)}$
    $\tau \leftarrow \tau + 1$
until $(|J^{(\tau)} - J^{(\tau-1)}| < \epsilon)$
return $\mathbf{R}$ (document assignment matrix), $\mathbf{W}$ (term discrimination info matrix)
```

4.1 Data Sets

Our experiments are conducted on 10 standard text data sets of different sizes, contexts, and complexities. The key characteristics of these data sets are given in Table 1. Data set 1 is obtained from the Internet Content Filtering Group's web site[1], data set 2 is available from a Cornell University web page[2], and data sets 3 to 10 are obtained from Karypis Lab, University of Minnesota[3]. Data sets 1 (stopword removal) and 3 to 10 (stopword removal and stemming) are available in preprocessed formats, while we perform stopword removal and stemming of data set 2. For more details on these standard data sets, please refer to the links given above.

4.2 Comparison Methods

We compare CDIM with five clustering methods. Four of them are K-means variants and one of them is based on Non-Negative Matrix Factorization (NMF) [8].

The four K-means variants are selected from the CLUTO Toolkit [18] based on their strong performances reported in the literature [5,3]. Two of them are direct K-way clustering methods while the remaining two are repeated bisection methods. For

[1] http://labs-repos.iit.demokritos.gr/skel/i-config/downloads/

[2] http://www.cs.cornell.edu/People/pabo/movie-review-data/

[3] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download

Table 1. Data sets and their characteristics

#	Name	Documents (N)	Terms (M)	Categories (K)
1	pu	672	19868	2
2	movie	1200	38408	2
3	reviews	4069	23220	5
4	hitech	2301	13170	6
5	tr31	927	10128	7
6	tr41	878	7454	10
7	ohscal	11162	11465	10
8	re0	1504	2886	13
9	wap	1560	8460	20
10	re1	1657	3758	25

each of these two types of methods, we consider two different objective functions. One objective function maximizes the sum of similarities between documents and their cluster mean. The direct and repeated bisection methods that use this objective function are identified as Direct-I2 and RB-I2, respectively. The second objective function that we consider maximizes the ratio of I2 and E1, where I2 is the intrinsic (based on cluster cohesion) objective function defined above and E1 is an extrinsic (based on separation) function that minimizes the sum of the normalized pairwise similarities of documents within clusters with the rest of the documents. The direct and repeated bisection methods that use this hybrid objective function are identified as Direct-H2 and RB-H2, respectively.

For NMF, we use the implementation provided in the DTU:Toolbox[4]. Specifically, we use the multiplicative update rule with Euclidean measure for approximating the term-document matrix.

In using the four K-means variants, the term-document matrix is defined by term-frequency-inverse-document-frequency (TF-IDF) values and the cosine similarity measure is adopted for document comparisons. For NMF, the term-document matrix is defined by term frequency values.

4.3 Clustering Validation Measures

We evaluate clustering quality with the BCubed metric [19]. In [20], it has been shown that the BCubed precision and recall are the only measures that satisfy all desirable constraints for a good clustering validation measure.

The BCUbed F-measure is computed as follows. Let $L(o)$ and $C(o)$ be the category and cluster of an object o. Then, the correctness of the relation between objects o and o' in the clustering is equal to one, $Correct(o, o') = 1$, iff $L(o) = L(o') \leftrightarrow C(o) = C(o')$; otherwise $Correct(o, o') = 0$. BCubed precision (BP) and BCubed recall (BR) can now be defined as: $BP = Avg_o[Avg_{o'.C(o)=C(o')}[Correct(o, o')]]$ and $BR = Avg_o[Avg_{o'.L(o)=L(o')}[Correct(o, o')]]$. The BCubed F-measure is then given by $BF = 2 \times \frac{BP \times BR}{BP+BR}$. The BCubed F-measure ($BF$) ranges from 0 to 1 with larger values signifying better clusterings.

[4] http://cogsys.imm.dtu.dk/toolbox/

5 Results and Discussion

5.1 Clustering Quality

Table 2 gives the results of the clustering quality evaluation. The desired number of clusters K for each data set is set equal to the number of categories in that data set (see Table 1). The shown values are average BCubed F-measure ± standard deviation, computed from 10 generated clusterings starting with random initial partitions.

These results show that CDIM outperforms the other algorithms on the ten data sets with five highest performance scores (shown in bold) and within 0.005 of the highest scores on three more data sets. CDIM is much better than NMF while its performances are closer to those of the K-means variants. We verified the consistency of these results using the Freidman's test, which is a non-parametric test recommended for evaluating multiple algorithms on multiple data sets [21]. At 0.05 significance level, CDIM is found to be significantly better than Direct-H2, RB-H2, and NMF, while its performance difference with Direct-I2 and RB-I2 is not statistically significant at this level.

An observation from our analysis is that CDIM consistently produces higher quality clusterings when the desired number of clusters is small (e.g. $K < 5$). This is attributable to the lesser resolution power of the multi-way comparisons ($\hat{d}_{ik} = d_{ik} - \bar{d}_{ik}, \forall k$) that are required for document assignment. One potential way to overcome this shortcoming for larger number of clusters is to use a repeated bisection approach rather than a direct K-way partitioning approach.

Table 2. Clustering quality evaluation (average BCubed F-measure ± standard deviation)

Data	**CDIM**	**Direct-I2**	**Direct-H2**	**RB-I2**	**RB-H2**	**NMF**
pu	**0.706±0.06**	0.565±0.02	0.553±0.02	0.565±0.02	0.553±0.02	0.612±0.04
movie	**0.581±0.02**	0.533±0.02	0.522±0.01	0.533±0.02	0.522±0.01	0.510±0.01
reviews	0.667±0.05	0.627±0.06	0.626±0.06	0.609±0.04	**0.669±0.03**	0.552±0.03
hitech	**0.433±0.04**	0.391±0.02	0.380±0.02	0.394±0.02	0.390±0.03	0.399±0.02
tr31	**0.636±0.11**	0.585±0.05	0.575±0.05	0.553±0.07	0.572±0.05	0.362±0.03
tr41	0.603±0.05	**0.608±0.02**	0.584±0.03	0.602±0.05	0.590±0.04	0.361±0.04
ohscal	0.429±0.02	0.422±0.02	0.417±0.03	**0.432±0.01**	0.427±0.01	0.250±0.02
re0	**0.417±0.02**	0.382±0.02	0.382±0.01	0.397±0.03	0.375±0.01	0.345±0.02
wap	0.442±0.05	0.462±0.01	0.444±0.01	**0.465±0.02**	0.438±0.02	0.299±0.02
re1	0.393±0.03	**0.443±0.02**	0.436±0.02	0.416±0.01	0.418±0.03	0.301±0.03

5.2 Cluster Understanding and Visualization

A key application of data clustering is corpus understanding. In the case of document clustering, it is important that clustering methods output information that can readily be used to interpret the clusters and their documents. CDIM is based on term discrimination information and each of its cluster is describable by the highly discriminating terms in it. We illustrate the understanding provided by CDIM's output by displaying

Table 3. Top 10 most discriminating terms (stemmed words) for clusters in ohscal data set

k	**Top 10 terms in cluster** k
1	‘platelet’, ‘kg’, ‘mg’, ‘dose’, ‘min’, ‘plasma’, ‘pressur’, ‘flow’, ‘microgram’, ‘antagonist’
2	‘carcinoma’, ‘tumor’, ‘cancer’, ‘surviv’, ‘chemotherapi’, ‘stage’, ‘recurr’, ‘malign’, ‘resect’, ‘therapi’
3	‘antibodi’, ‘antigen’, ‘viru’, ‘anti’, ‘infect’, ‘hiv’, ‘monoclon’, ‘ig’, ‘immun’, ‘sera’
4	‘patient’, ‘complic’, ‘surgeri’, ‘ventricular’, ‘infarct’, ‘oper’, ‘eye’, ‘coronari’, ‘cardiac’, ‘morta’
5	‘pregnanc’, ‘fetal’, ‘gestat’, ‘matern’, ‘women’, ‘infant’, ‘deliveri’, ‘birth’, ‘labor’, ‘pregnant’
6	‘risk’, ‘alcohol’, ‘age’, ‘children’, ‘cholesterol’, ‘health’, ‘factor’, ‘women’, ‘preval’, ‘popul’
7	‘gene’, ‘sequenc’, ‘dna’, ‘mutat’, ‘protein’, ‘chromosom’, ‘transcript’, ‘rna’, ‘amino’, ‘structur’
8	‘contract’, ‘muscle’, ‘relax’, ‘microm’, ‘calcium’, ‘effect’, ‘respons’, ‘antagonist’, ‘releas’, ‘action’
9	‘il’, ‘receptor’, ‘cell’, ‘stimul’, ‘bind’, ‘growth’, ‘gamma’, ‘alpha’, ‘insulin’, ‘0’
10	‘ct’, ‘imag’, ‘comput’, ‘tomographi’, ‘scan’, ‘lesion’, ‘magnet’, ‘reson’, ‘cerebr’, ‘tomograph’

the top 10 most discriminating terms (stemmed words) for each cluster of the ohscal data set in Table 3. The ohscal data set contains publications from 10 different medical subject areas (antibodies, carcinoma, DNA, in-vitro, molecular sequence data, pregnancy, prognosis, receptors, risk factors, and tomography). By looking at the top ten terms, it is easy to determine the category of most clusters: cluster 2 = carcinoma, cluster 3 = antibodies, cluster 4 = prognosis, cluster 5 = pregnancy, cluster 6 = risk factors, cluster 7 = DNA, cluster 9 = receptors, cluster 10 = tomography. The categories molecular sequence data and in-vitro do not appear to have a well-defined cluster; molecular sequence data has some overlap with cluster 7 while in-vitro has some overlap with clusters 1 and 9. Nonetheless, clusters 2 and 8 still give coherent meaning to the documents they contain.

As another example, in hitech data set, the top 5 terms for two clusters are: (1) ‘health’, ‘care’, ‘patient’, ‘hospit’, ‘medic’, and (2) ‘citi’, ‘council’, ‘project’, ‘build’, ‘water’. The first cluster can be mapped to the health category while the second cluster does not have an unambiguous mapping to a category but it still gives sufficient indication that these articles discuss hi-tech related development projects.

Since CDIM finds clusters in a K-dimensional discrimination information space, the distribution of documents among clusters can be visualized via simple scatter plots. The 2-dimensional scatter plot of documents in the pu data set is shown in Figure 1 (left plot). The x- and y-axes in this plot correspond to document discrimination information for cluster 1 and 2 (d_{i1} and d_{i2}), respectively. and the colored makers give the true categories. It is seen that the two clusters are spread along the two axes and the vast majority of documents in each cluster belong to the same category. Similar scatter plots for Direct-I2 and NMF are shown in the middle and right plots, respectively, of Figure 1. However, these methods exhibit poor separation between the two categories in the pu data set.

Such scatter plots can be viewed for any pair of clusters when $K > 2$. Since CDIM's document assignment decision is based upon document discrimination scores ($\hat{d}_{ik}, \forall k$), scatter plots of documents in this space are also informative; each axis quantifies how relevant a document is to a cluster in comparison to the remaining clusters.

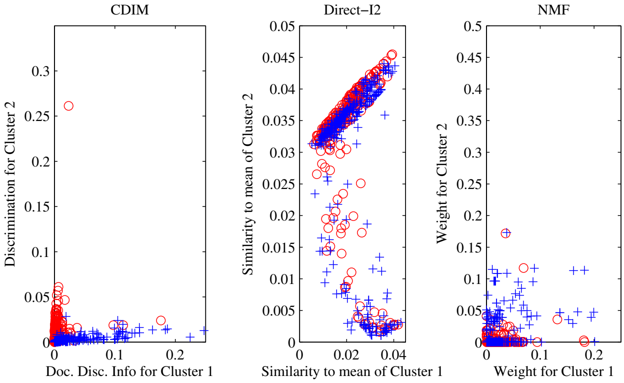


Fig. 1. Scatter plot of documents projected onto the 2-D discrimination information space (CDIM), similarity to cluster mean space (Direct-I2), and weight space (NMF). True labels are indicated by different color markers.

6 Conclusion and Future Work

In this paper, we propose and evaluate a new document clustering method, CDIM, that finds clusters in a K-dimensional space in which documents are well discriminated. It does this by maximizing the sum of the discrimination information provided by documents for their respective clusters minus that provided for the remaining clusters. Document discrimination information is computed from the discrimination information provided by the terms in it. Term discrimination information is estimated from the document collection via its relative risk. An advantage of using a measure of discrimination information is that it also quantifies the degree of relatedness of a term to its context in the collection. Thus, CDIM produces clusters that are readily interpretable by their highly discriminating terms.

Our experimental evaluations confirm the effectiveness of CDIM as a practically useful document clustering method. Its core idea of clustering in spaces defined by corpus-based discrimination or relatedness information holds much potential for future extensions and improvements. In particular, we would like to investigate other measures of discrimination/relatedness information, extend and evaluate CDIM for soft clustering, and develop a hierarchical and repeated bisection version of CDIM.

References

1. Morris, J., Hirst, G.: Non-classical lexical semantic relations. In: Proceedings of the HLT-NAACL Workshop on Computational Lexical Semantics, pp. 46–51. Association for Computational Linguistics (2004)
2. Cai, D., van Rijsbergen, C.J.: Learning semantic relatedness from term discrimination information. Expert Systems with Applications 36, 1860–1875 (2009)
3. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison Wesley, New York (2006)

4. Zhao, Y., Karypis, G.: Criterion functions for document clustering: experiments and analysis. Technical Report 01-40, University of Minnestoa (2001)
5. Steinbach, M., Karypis, G.: A comparison of document clustering techniques. In: Proceedings of the KDD Workshop on Text Mining (2000)
6. Hu, X., Zhang, X., Lu, C., Park, E., Zhou, X.: Exploiting wikipedia as external knowledge for document clustering. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 389–396. ACM (2009)
7. Zhang, X., Jing, L., Hu, X., Ng, M., Zhou, X.: A Comparative Study of Ontology Based Term Similarity Measures on PubMed Document Clustering. In: Kotagiri, R., Radha Krishna, P., Mohania, M., Nantajeewarawat, E. (eds.) DASFAA 2007. LNCS, vol. 4443, pp. 115–126. Springer, Heidelberg (2007)
8. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, pp. 267–273. ACM (2003)
9. Xu, W., Gong, Y.: Document clustering by concept factoriz ation. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 202–209. ACM (2004)
10. Cai, D., He, X., Han, J.: Locally consistent concept factorization for document clustering. IEEE Transactions on Knowledge and Data Engineering (2010)
11. Tang, B., Shepherd, M., Heywood, M.I., Luo, X.: Comparing Dimension Reduction Techniques for Document Clustering. In: Kégl, B., Lee, H.-H. (eds.) Canadian AI 2005. LNCS (LNAI), vol. 3501, pp. 292–296. Springer, Heidelberg (2005)
12. Ding, C., Li, T.: Adaptive dimension reduction using discriminant analysis and k-means clustering. In: Proceedings of the 24th International Conference on Machine Learning, pp. 521–528. ACM (2007)
13. Junejo, K., Karim, A.: A robust discriminative term weighting based linear discriminant method for text classification. In: Eighth IEEE International Conference on Data Mining, pp. 323–332 (2008)
14. Li, H., Li, J., Wong, L., Feng, M., Tan, Y.P.: Relative risk and odds ratio: a data mining perspective. In: PODS 2005: Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (2005)
15. Li, J., Liu, G., Wong, L.: Mining statistically important equivalence classes and delta-discriminative emerging patterns. In: KDD 2007: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2007)
16. Hsieh, D.A., Manski, C.F., McFadden, D.: Estimation of response probabilities from augmented retrospective observations. Journal of the American Statistical Association 80(391), 651–662 (1985)
17. LeBlanc, M., Crowley, J.: Relative risk trees for censored survival data. Biometrics 48(2), 411–425 (1992)
18. Karypis, G.: CLUTO-a clustering toolkit. Technical report, Dept. of Computer Science, University of Minnesota, Minneapolis (2002)
19. Bagga, A., Baldwin, B.: Entity-based cross-document coreferencing using the vector space model. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, vol. 1, pp. 79–85. ACL (1998)
20. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval 12(4), 461–486 (2009)
21. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)

A Semi-supervised Incremental Clustering Algorithm for Streaming Data

Maria Halkidi[1], Myra Spiliopoulou[2], and Aikaterini Pavlou[3]

[1] Dept of Digital Systems, University of Piraeus
mhalk@unipi.gr
[2] Faculty of Computer Science, Magdeburg University, Germany
myra@iti.cs.uni-magdeburg.de
[3] Athens University of Economics and Business
aikaterinipavlou@gmail.com

Abstract. Nowadays many applications need to deal with *evolving data streams*. In this work, we propose an incremental clustering approach for the exploitation of user constraints on data streams. Conventional constraints do not make sense on streaming data, so we extend the classic notion of constraint set into a *constraint stream*. We propose methods for using the constraint stream as data items are forgotten or new items arrive. Also we present an on-line clustering approach for the cost-based enforcement of the constraints during cluster adaptation on evolving data streams. Our method introduces the concept of multi-clusters (m-clusters) to capture arbitrarily shaped clusters. An m-cluster consists of multiple dense overlapping regions, named s-clusters, each of which can be efficiently represented by a single point. Also it proposes the definition of outliers clusters in order to handle outliers while it provides methods to observe changes in structure of clusters as data evolves.

Keywords: stream clustering, semi-supervised learning, constraint-based clustering.

1 Introduction

Clustering plays a key role in data analysis, aiming at discovering interesting data distributions and patterns in data. Also it is widely recognized as means to provide an effective way of maintaining data summaries and a useful approach for outlier analysis. Thus clustering is especially important for streaming data management. Streaming data are generated continuously at high rates and due to storage constraints we are not able to maintain in memory the entire data stream. Having a compressed version (synopsis) of data at different time slots, data analysts are able to keep track of the previously arrived data and thus more effectively extract useful patterns from the whole stream of data.

Semi-supervised clustering has received much attention in the last years, because it can enhance clustering quality by exploiting readily available background knowledge, i.e. knowledge on the group membership of some data items or knowledge on some properties of the clusters to be built. The available knowledge is

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 578–590, 2012.

represented in form of constraints (must-link and/or cannot-link constraints) which are incorporating in the clustering procedure. There is much research on semi-supervised clustering for static data [5,4,9,7,8,15], but less for *streaming data* [13]. However a huge amount of data generated in everyday life can be characterized as data streams (e.g. sensor data, web logs, Internet traffic, RFID) while many of the applications that profit from semi-supervised clustering, are essentially analyzing streaming data. For instance, consider a stream of news in document form and an application that clusters documents in topics. When a piece of news comes in as part of an already existing story, some news agencies also provide links to previous documents of the story. These links can be considered as constraints that indicate which documents are correlated and thus should be classified together.

Semi-supervised stream clustering requires an extension of the successful concept of *instance-level constraints*: Any record in a stream is gradually forgotten; if it is involved in a constraint, then the constraint itself has a limited lifetime. At the same time, some of the arriving items may be labeled, implying new constraints. We deal with both issues in our `Semi`-supervised `Stream` clustering method (further referred to as `SemiStream`). According to our approach a clustering scheme is adapted to a continuous flow of items, some of which carry labels and thus give raise to instance-level constraints. These labeled data are mapped into Must-Link and Cannot-Link constraints, thus forming a *constraint stream* that accompanies the data stream. Adaptation encompasses the elimination of old items and outdated constraints and the adjustment of the clusters to newly arriving items, taking account of all to-date constraints.

`SemiStream` starts with an initial clustering based on a given set of constraints. Then, at each period of observation t_i, we adapt the clusters to accommodate new data and satisfy new constraints, while old data are gradually forgotten and obsolete constraints (on forgotten data) are eliminated. To this purpose, we propose a *constraint-cost function* that associates constraint violations with penalty values. We use this function to assign each new data item to the cluster with the most proximal representative and incurring minimal cost. We specify an upper boundary to the cost that can be tolerated during cluster adaptation. Then, data items that cannot be placed to clusters without exceeding this boundary are declared as *outliers* and are accommodated to *outlier groups*. Finally, we identify and merge overlapping clusters, thereby checking the value of the constraint-cost function.

To summarize, the contributions of our work are as follows:

1) We propose an incremental approach for clustering evolving data streams based on constraints. The notion of constraint stream is introduced so that we efficiently describe the sequential fashion of labeled data that may emerge as data flows.

2) Our approach adopts the use of i) multiple clusters to represent significant neighboring dense areas in data, capturing thus arbitrarily shaped clusters, and ii) outliers clusters to describe a small set of data whose characteristics seem to deviate significantly from average behavior of the currently processed data.

2 Related Work

Recently, a number of clustering algorithms have been proposed to deal with streaming data. In [1], a framework of a stream clustering approach is proposed, which includes two clustering phases. At the online phase, the proposed approach periodically maintain statistical information about local data in terms of *micro-clusters*, while at the off-line phase, the decision maker uses these statistics to provide a description of clusters in the data stream. The main drawback of this algorithm is that the number of micro-clusters needs to be predefined. HPStream [2] incorporates a fading cluster structure and a projection-based clustering methodology to deal with the problem of high-dimensionality in data streams. Another stream clustering algorithm is Denstream [6]. It is based on the idea of DBSCAN [10] and it forms local clusters progressively by detecting and connecting dense data item neighborhoods.

A version of the AP algorithm that is closer to handling streaming data is presented in [16]. According to this approach, as data flows, data items are compared one-by-one to the exemplars and they are assigned to their nearest one if a distance threshold condition is satisfied. Otherwise, data are considered outliers and they are put in a reservoir. A cluster redefinition is triggered if the number of outliers exceeds a heuristic (user-defined) reservoir size or if a change in data distribution is detected.

Though there is a lot of work on constraint-based clustering methods for static data [3,8,15], the related work on clustering streaming data based on constraints is limited. Ruiz et al. [14] have presented a conceptual model for constraints on streams and extended the constraint-based K-means of [15] for streaming data. In SemiStream, we refine this model by proposing a *constraint stream* of instance-level constraints and we adapt the clusters incrementally instead of rebuilding them from scratch. Moreover an extension of Denstream so that constraints are taken into account during the clustering process is proposed in [13]. C-Denstream ensures that all given constraints are satisfied. However this is achieved at the cost of creating many small clusters. Moreover, in case of conflicts among constraints, C-Denstream is unable to conclude in a clustering.

3 A Model for Constraint-Based Clustering on Streaming Data

We study a stream of items, for some of which we have background knowledge in the form of instance-level constraints. We model constraints and clusters over a stream and then model the cost of assigning an item to a cluster, thereby possibly violating some constraints.

3.1 Modeling a Stream of Constraints

Let $t_1, \ldots, t_n$ be a series of time points. The stream is partitioned in data snapshots $D_1, \ldots, D_n$, where D_i consists of the items arriving during $(t_{i-1}, t_i]$. The

Algorithm 1. DYNAMIC CONSTRAINT UPDATER$(D, CS, t, \widehat{CS}_{old})$

```
Data: Snapshot $D$ at $t$, new constraint-set $CS$, set of active constraints $\widehat{CS}_{old}$.
begin
    $\widehat{CS} = CS \cup \widehat{CS}_{old}.$ ;
    for $cs \in \widehat{CS}$ do
    ;
    Let $cs = \prec x, y \succ$;
    if $teenage(x, t) == 0$ or $teenage(y, t) == 0$ then $\widehat{CS}_i = \widehat{CS} \setminus \{cs\}.$ ;
    else $weight(cs) = \min\{teenage(x, t), teenage(y, t)\}.$ ;
    return $\widehat{CS}$.;
end
```

dataset remembered at each t_i, $\widehat{D}_i \subseteq \cup_{j=1}^{i} D_j$ is determined by a decay age function. Presently, we assume a sliding window of size $w \in [0, n]$, so that $\widehat{D}_i = \cup_{j=u}^{i} D_j$ for $u = \max\{1, i - w + 1\}$. We further assign weights to the items inside the window, so that more recent items acquire higher weights. So, at time point t_i, the weight of an item x that arrived at time point $t_j \leq t_i$ is $teenage(x, t_i) = 1 - \frac{t_i - t_j}{w}$.

We consider Must-Link and Cannot-Link instance-level constraints [15]. Must-link constraints indicate data items that should belong to the same cluster while Cannot-link constraints refer to data items that should not be assigned to the same cluster. Such constraints involve two items x, y and are denoted as $\prec x, y \succ$. Since constraints refer to items that are associated with an age-dependent weight, they also have weights. So, at t_i, the weight of a constraint $cs = \prec x, y \succ$ is

$$weight(cs, t_i) = \min\{teenage(x, t_i), teenage(y, t_i)\} \tag{1}$$

At each time point t_i, some items of the snapshot D_i are involved in instance-level constraints, together with items of earlier time points. These new constraints constitute the *new* constraint-set CS_i. From them, we derive the set of *active constraints* at t_i, $\widehat{CS}_i = \cup_{j=u}^{i} CS_j, u = \max\{1, i - w\}$. In Alg. 1, we depict the DYNAMIC CONSTRAINT UPDATER, which builds $\widehat{CS}_i$. At each $t = t_2, t_3, \ldots$, the DYNAMIC CONSTRAINT UPDATER reads the set of old active constraints $\widehat{CS}_{old}$ and the current constraint-set CS. It combines them to produce the new set of active constraints $\widehat{CS}$ by (a) marking constraints on outdated items as *obsolete*, (b) recomputing the weights of non-obsolete, i.e. *active* constraints and (c) taking the union of the resulting set of constraints with CS to form $\widehat{CS}$.

3.2 Cost of Assigning Data Items to a Cluster

The assignment of an item x to a cluster c at time point t incurs a *cost*, which reflects constraint violations caused by the assignment, as well as the distance of x to the center of c. We first model the cost of constraint violations. Let $\widehat{CS}$ be the active set of constraints. From this set, we derive $ML(x)$, the set of items that are involved in Must-Link constraints together with x, and $CL(x)$, the corresponding set of items for Cannot-Link constraints. Then, the constraint-violation cost for assigning x to c is given by:

$$\begin{aligned} &costCV(x, c, \widehat{CS}, t) = \textstyle\sum_{y \in ML(x), y \notin c} weight(\prec x, y \succ, t) \times f_{ML}(x, y) + \\ &\textstyle\sum_{y \in CL(x), y \in c} weight(\prec x, y \succ, t) \times f_{CL}(x, y) \end{aligned} \qquad (2)$$

In this formula, we consider the items that must be linked with x and are not members of cluster c, as well as the items in c that should not be linked to x. For such an item y, the weight of the corresponding constraint is $weight(\prec x, y \succ, t)$, according to Eq. 1.

The functions $f_{ML}(\cdot)$, $f_{CL}(\cdot)$ denote the cost of violating a Must-Link, resp. Cannot-Link constraint. We consider that the cost of violating a must-link constraint between two close points should be higher than the cost of violating a must-link constraint between two points that are far apart. Thus we implement the cost function $f_{ML}(\cdot)$ as $f_{ML}(x, y) = d_{max} - d(x, y)$, where d_{max} is the maximum distance encountered between two items in the dataset. Similarly, the cost of violating a cannot-link constraint between two distant points should be higher than the cost of violating a cannot-link constraint between points that are close. Then we define the $f_{CL}(\cdot)$ function as the distance between the two items involved in a cannot-link constraint and we set $f_{CL}(x, y) = d(x, y)$.

Then, the cost of assigning an item x to a cluster c for a given set of active constraints $\widehat{CS}$ at time point t consists of the cost of effected constraint violations and the overhead of placing this item to this cluster. The latter is represented by the distance of the item to the cluster center:

$$cost(x, c, \widehat{CS}, t) = costCV(x, c, \widehat{CS}, t) + d(x, rep(c)) \qquad (3)$$

3.3 A Clustering Model for Streaming Data

Our constraint-based clustering approach is based on the incremental adjustment of the clusters to the arrival of new data and constraints and to the decay of old data and constraints. In this approach, we distinguish between items that can be assigned to a cluster and those that cannot, either because they violate some constraints or because they are far away from any cluster. We call the latter *outlier items*. As the stream of data and the stream of constraints proceed, it may become possible to merge earlier clusters together and even place outlier items to some cluster. We present here a model that captures those cases, by distinguishing between *s-clusters* which correspond to core clusters of data, *o-clusters* that are groups of outlier items and *m-clusters* that are the result of merging core data clusters, i.e. dense areas in the dataset.

An *s-cluster* is a conventional cluster c, built at time point t_i by a constraint-based clustering algorithm like MPCK-Means. For such a cluster, we define concepts that describe its structure and surroundings.

Definition 1 (s-cluster Nucleus). *Let c be an s-cluster. We denote as $rep(c)$ the representative or conceptual center of c, consisting of the mean values of the* *items in c across all dimensions of the feature space. The within-cluster distance of c is the average distance of a data item from the cluster representative:*

$$withinClusterDistance(c) = \frac{\sum_{x \in c} d(x, center(c))}{|c|} \tag{4}$$

With some abuse of conventions, we also use the term "radius" for the within-cluster-distance, i.e. we set $radius(c) \equiv withinClusterDistance(c)$*. Then, the items within the radius of c constitute its "nucleus":*

$$nucleus(c) = \{x \in c | d(x, center(c) \leq radius(c)\} \tag{5}$$

By Def. 1, we distinguish between items close to the s-cluster's representative (center) and remote ones. Pictorially, the former constitute the core or nucleus of the s-cluster, while the latter form the rim of the s-cluster. Also data items that are distant from the rest data are perceived as outliers.

Definition 2 (s-cluster Rim). *Let c be a s-cluster. Assume that* ξ *is the current clustering of streaming data. Then, the* rim *of c consists of the data items that are outside the nucleus of c but their distance from the representative of c is lower than some multiple* $\tau > 1$ *of its radius:*

$$rim(c) = \{x \in c | radius(c) < d(x, rep(c)) \leq \tau \times radius(c)\}$$

whereby depending on the specification of τ*, a cluster may have an empty rim.*

Next to items that are too far from the center of their s-cluster, we have also items whose assignment violates an instance-level constraint. For these data items, we use the term *outlier (items)*.

Definition 3 (Outlier Item). *Assume that* ξ *is a clustering of streaming data that has been defined at time point t based on a given set of active constraints* $\widehat{CS}$*. We say that a data item x is an "outlier" for a s-cluster* $c \in \xi$ *or, equivalently, that x belongs to the set* $outliers(c, CS, t)$*, if x is closest to c than any other cluster in* ξ*, but*
1) x does not belong to the rim of c (see Def. 2), and
2) the cost of assigning x to c with respect to $\widehat{CS}$ *exceeds a threshold* τ_{CS}*, i.e.* $costCV(x, c, \widehat{CS}, t) > \tau_{CS}$

The threshold τ_{CS} *is user-defined and indicates our tolerance to constraint violation. Thus if* $\tau_{CS} = 0$ *then none of the given constraints are allowed to be violated.*

At time point t, the items of a data batch are assigned to the existing s-clusters. After assigning all items, the clusters' nuclei, rims and outliers are recomputed. It may then happen that two clusters have moved closer to each other so that their rims overlap. If their nuclei also overlap, then they are candidates for merging.

Definition 4 (Overlap between s-clusters). *Let c_1, c_2 be two s-clusters built at time point t. The* overlap *of c_2 with respect to c_1 is the number of items in the nucleus of c_2 that are within the nucleus of c_1, normalized to the cardinality of c_1:*

$$overlap(c_1, c_2) = \frac{1}{|c_1|} \times |\{x \in nucleus(c_2) | d(x, rep(c_1)) \leq radius(c_1)\}| \qquad (6)$$

The $overlap(c_2, c_1)$ is defined accordingly. We say that the s-clusters c_1, c_2 "overlap" if either $overlap(c_1, c_2)$ or $overlap(c_2, c_1)$ is larger than some threshold $\tau_{overlap}$.

When two nuclei overlap, the s-clusters are so close that they can be merged into one cluster, which we term as *m-cluster.*

Definition 5 (m-cluster). *Let ξ be the set of s-clusters built at time point t. A "multi-cluster" or "m-cluster" $C \subseteq \xi$ is a group of s-clusters, such that:*
1)For each s-cluster $c \in C$ there is a s-cluster $c' \in C$ such that c, c' overlap according to Def. 4.
2) For each s-cluster $c \in \xi$ such that there is a s-cluster $c' \in C$ that overlaps with c it holds that $c \in C$.

Then we define the representatives of a m-cluster C as the set of its s-clusters' representatives, i.e. $rep(C) = \{rep(c) | c \in C\}$.

By this definition, a m-cluster is a maximal set of overlapping s-clusters within the clustering built at a certain time point. The reader should notice the similarity to the definition of "cluster" in DBSCAN [10] as a maximal group of overlapping neighborhoods. Our notion of "overlap" is the counterpart of density-connectivity as proposed in [10]. The advantage of defining m-clusters for constraint-based stream mining is that an algorithm like DBSCAN lends itself elegantly to constraint enforcement, as has been shown in [12].

Definition 6 (o-cluster). *Let t_i be a time point and ξ be the set of s-clusters built at this time point. A group of (constraint-based) outlier items c_o constitutes an "outlier cluster" or "o-cluster" if and only if:*
1) For each $x, y \in c_o$ and for each $c \in \xi$ it holds that $d(x, y) < d(x, rep(c))$ and $d(x, y) < d(y, rep(c))$.
2)For each $x, y \in c_o$ there is no Cannot-Link constraint $< x, y >$.

Similarly to a s-cluster, an o-cluster c_o has a representative $rep(c_o)$ composed of the mean of the elements of c_o. However, we define neither nucleus nor rim for an o-cluster, since the assignment of items to it is not driven by proximity to the o-cluster's center but by remoteness to neighboring s-clusters.

The reader may have noticed that the above definition contains no specification of maximality. In the next section, we explain how o-clusters are built progressively and compensate for the absence of a maximality constraint here.

4 Incremental Clustering towards a Constraint-Stream

Our incremental clusterer `SemiStream` takes as input a stream of data items arriving in snapshots/batches $D_1, \ldots, D_n$, an accompanying stream of arriving constraints $CS_1, \ldots, CS_n$ and the cost function of Eq. 3 for the assignment of items to clusters, subject to Must-Link and Cannot-Link constraints.

At the beginning of the observation time t_0, we assume an initial constraint-based clustering with a conventional semi-supervised algorithm like MPCK-Means [4]. At each subsequent time point t_i, we assign all items of the batch D_i into s-clusters, re-compute the s-clusters' nuclei, their rims and outlier items. Then, we study the proximity of outliers to each other, eventually merging them into o-clusters. Similarly, we check the proximity of s-clusters to their surroundings and merge them into m-clusters - or detach them accordingly.

Processing an Item of the Stream. Let D_i be the set of data items arriving at $(t_{i-1}, t_i]$ and let ξ be the set of s-level clusters at this time point, where some s-clusters may be part of an earlier formed m-cluster. For each item $x \in D_i$ we consider the following cases:

1. There is a s-cluster $c \in \xi$ so that $x \in \mathit{nucleus}(c)$ *and* $\mathit{cost}(x, c, CS, t_i) < \tau_{CS}$: x is assigned to c.
2. Otherwise, x is termed as outlier.

Building Outlier Clusters. If an item is defined to be an outlier, we check whether there is another item that is proximal to it and with which it can be grouped without violating any Cannot-Link constraints. The two items become part of an *outlier cluster*.

When computing the proximity of an item to an s-cluster, we do so on the basis of the cluster's nucleus and radius. When considering the proximity of an outlier to another outlier, we do not have such a basis. We therefore set a constant ε and specify that an item x can be assigned to an outlier cluster c_o if and only if $d(x, \mathit{rep}(c_o) < \varepsilon$ and there is no violation constraint. The center of an outlier cluster is defined as the mean of the points that belong to this cluster. This allows us to group items together into an outlier cluster without allowing the *withinClusterDistance* of the outlier cluster to grow exuberantly (cf. Def. 1).

The constant ε can be set to a small value, e.g. equal to the smallest nucleus encountered at t_i, thus allowing only groups of very proximal outliers. Alternatively, ε may be set to a large value, e.g. equal to the largest encountered nucleus, thus allowing rather distant outliers to form a cluster. In any case, when specifying ε at each time point t_i, it must be checked whether the constant will allow an overlap between the outlier cluster and one of the s-clusters. In that case, the clusters can be merged.

Moreover, we note that *an outlier cluster can grow into a s-cluster*. We consider only o-clusters that are adequately large and homogeneous:

- An o-cluster is *adequately large* if its cardinality is comparable to the cardinality of the smallest s-cluster.

– An o-cluster is *adequately homogeneous* if its variance is comparable to the average variance of s-clusters.

Candidates for Cluster Merging. After incorporating all items of D_i into the original clustering (i.e. previously defined clustering ξ), some clusters may have grown or moved closer to each other. If their nuclei have come to overlap, we can merge them into an m-cluster. At the same time, some clusters may have moved apart from each other; if they were previously part of an m-cluster, then we must detach them. In general, we consider the following types of candidate clusters:

1. c, c' are *s-clusters*, such that $overlap(c, c') > \tau_{overlap}$ or $overlap(c', c) > \tau_{overlap}$.
2. c, c' are *o-clusters*, such that $d((rep(c), rep(c')) < \varepsilon$; an o-cluster may consist of a single item.

For each pair of candidates we check whether there is a Cannot-Link constraint $< x, y >$ such that $x \in c, y \in c'$. If $cost(x, c', CS) < \tau_{CS}$ (or $cost(y, c, CS) < \tau_{CS}$), then the clusters can be merged. We merge s-clusters into an m-cluster, so that nuclei and rims become part of the m-cluster.

As a third case, we may consider the *merging of an s-cluster with an o-cluster* into an m-cluster, if the center of the o-cluster is within the nucleus of the s-cluster and no Cannot-Link constraints are violated by the merge. However, we consider only o-clusters that can grow into s-clusters, i.e. are adequately large and homogeneous.

In the cases discussed above, the s-clusters may be already members of distinct m-clusters. If this holds true, then the test for constraint violation is extended to all members of each m-cluster involved.

The new clustering ξ', defined after applying any of the above merging cases to ξ, is evaluated based on clustering quality criteria (i.e. cluster compactness and separation). The new clustering is acceptable if its quality does not significantly changes with respect to the quality of ξ.

Candidates for M-Cluster Split. After considering the merging of clusters, we also consider cases where m-clusters need to be split. This may happen because of new Cannot-Link constraints and because of changes in the clusters' nuclei:

1. c is member of an m-cluster C and there is no $c' \in C$ such that $overlap(c, c') + overlap(c', c) > \tau_{overlap}$.
2. c is an o-cluster and member of an m-cluster C and there is no $c' \in C$ such that $rep(c) \in nucleus(c')$.
3. c is member of an m-cluster C and there is a Cannot-Link constraint such that $x \in c$ and $y \in C \setminus c$.

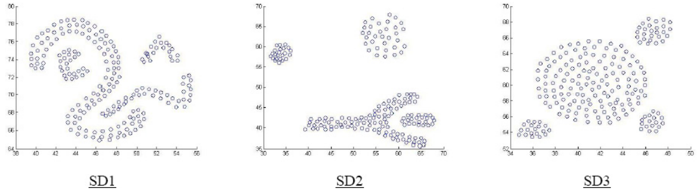
In all cases, the cluster is removed from the m-cluster, possibly causing further cluster detachments. Similarly to the merging case, the clustering defined after splitting is evaluated based widely known cluster quality criteria. Specifically, a cluster validity method is adopted to evaluate the results of the proposed incremental clustering approach at specific time slots in terms of constraint accuracy as well as clustering compactness and separation [11]. Then if the quality of the newly emerged clustering ξ' is comparable to the old clustering, ξ' is an accepted clustering. The cluster validity method may trigger the re-clustering of data when significant changes are observed in the quality of currently defined clusters.

5 Experimental Evaluation

In this section we present experimental evaluation of our approach using different datasets. We implemented `SemiStream` in JAVA. All experiments were conducted on a 2.53GHz Intel(R) Core 2 Duo PC with 6GB memory.

Data Sets and Constraints Generation. We generate some synthetic datasets with different numbers of clusters and dimensions. For the sake of visualization, we chose here to present the performance of our approach on two-dimensional datasets. Specifically, to evaluate the sensitivity of our algorithm and its clustering quality in case of arbitrarily shaped clusters, we consider the synthetic datasets depicted in Figure 1, which have also been used in [10]. We also generated an evolving data stream, ESD, by choosing one of the three datasets (denoted SD1, SD2, SD3 in Figure 1) 10 times. Each of the datasets contains $10,000$ points and thus the total length of ESD is $100,000$.

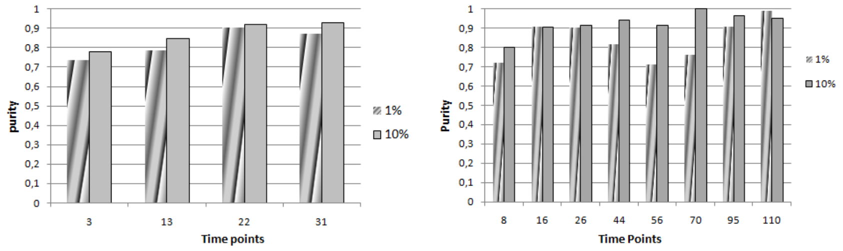
To generate constraints for our experimental study, we consider a set of labeled data. Our algorithm is fed with constraints each time a new batch of data arrive. Following a strategy similar to this used in the static semi-supervised approaches, the constraints are generated from labeled data so that they correspond to a percentage of data in the considered window.



SD1 SD2 SD3

Fig. 1. Visualization of synthetic datasets

Evaluation Criterion. Evaluation of quality of clustering amounts to measuring the purity of defined clusters with respect to the true class labels. For experimental purposes, we assume that we know the label of data items. We run our approach on the set of unlabeled data and we assess the purity of the defined clusters. Then we can define the purity measure as follows: $Purity = \frac{1}{k} \cdot \sum_{i=1}^{k} \frac{|C_i^d|}{|C_i|}$,



(a) (b)

Fig. 2. Clustering purity vs Time points: a) SD1 dataset, horizon = 2, stream speed = 250, b) ESD dataset, horizon = 2, stream speed = 1,000

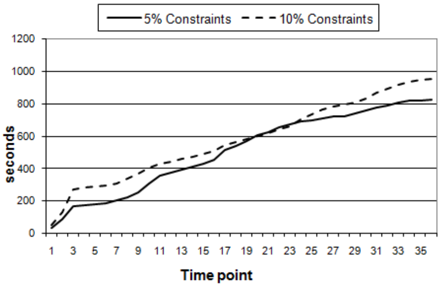


Fig. 3. Execution time vs length of stream

where $|C_i^d|$ denotes the number of majority class items in cluster i, $|C_i|$ is the size of cluster i and k is the number of clusters. The results of clustering purity presented in the following section are an average of results over 5 runs of our algorithm.

Clustering Quality Evaluation. First, we test the clustering quality of SemiStream using the SD1 dataset. We consider that the stream speed is 250 data items per time point, the window size is set to 4 time points. Also a new set of stream constraints is received as data arrives which corresponds to a percentage of data in the window. We can observe that the clusters are arbitrarily shaped and thus the majority of clustering algorithms are not able to identify them. Our study shows that the use of constraints can assist with the clustering procedure. Since the points fades out as time passes, we compute the purity of clustering results in a pre-defined horizon (h) from current time. Figure 2(a) presents the purity of clustering defined by SemiStream in a small horizon ($h = 2$) when the constraints are 1% and 10% of the arrived data. It can be seen that SemiStream gives very good clustering quality. The clustering purity is always higher than 75%. Also Figure 2 (a) shows that increasing the number of constraints, the purity of identified clusters is increased.

Then we evaluate the performance of our algorithm using the evolving data stream ESD at different time units. We set the stream speed at $1,000$ points per time unit and horizon equals to 2. Figure 2 (b) depicts the clustering purity results of our algorithm when the constraints correspond to 1% and 10% of the arrived data. It can be seen that SemiStream achieves to identify the evolution of the clusters as new data arrives, resulting in clusters with purity higher than 80%. Also we can observe the advantage that the use of constraints provides. Using 10% of data as constraints, our approach can achieve a clustering model with purity 99%.

Time Complexity. We evaluate the efficiency of SemiStream measuring the execution time. The algorithm periodically stores the current clustering results. Thus the execution time refers to the time that our algorithm needs to store clustering results, read data from previous time slots and redefine clusters. Figure 3 shows execution time for synthetic dataset using different number of constraints. We can observe that the execution time grows almost linearly as data stream proceeds.

6 Conclusions

We present SemiStream, an algorithm that incrementally adapts a clustering scheme to streaming data which are also accompanied by a set of constraints. Modeling constraints as a stream and associating constraint violation with a penalty function allowed us to design a cost-based strategy for cluster adaptation to snapshots of arriving data and constraints. We introduce the use of i) *s-clusters* to describe dense areas in the data set and ii) *multiple clusters (m-clusters)* to represent overlapping dense areas in order to capture arbitrarily shaped clusters. Moreover we use the structure of outliers clusters to describe a small set of data whose characteristics seem to deviate significantly from average behavior of the currently processed data. Based on a set of adaptation criteria SemiStream achieve to observe changes in structure of clusters as data evolve.

References

1. Aggarwal, C., Han, J., Wang, J., Yu, P.: A framework for clustering evolving data streams. In: Proc. of VLDB (2003)
2. Aggarwal, C., Han, J., Wang, J., Yu, P.: A framework for projected clustering of high dimensional data streams. In: Proc. of VLDB (2004)
3. Basu, S., Banerjee, A., Mooney, R.J.: Semi-supervised Clustering by Seeding. In: Proc. of ICML (2002)
4. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: Proc. of ICML (2004)
5. Bilenko, M., Basu, S., Mooney, R.J.: A probabilistic framework for semi-supervised clustering. In: Proc. of KDD, p. 8 (2004)
6. Cao, F., Ester, M., Qian, W., Zhou, A.: Density-based clustering over an evolving data stream with noise. In: Proc. of SDM (2006)
7. Davidson, I., Ravi, S.S.: Agglomerative Hierarchical Clustering with Constraints: Theoretical and Empirical Results. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 59–70. Springer, Heidelberg (2005)

8. Davidson, I., Ravi, S.: Clustering with constraints: Feasibility issues and the k-means algorithm. In: Proc. of SDM, Newport Beach, CA (April 2005)
9. Davidson, I., Wagstaff, K.L., Basu, S.: Measuring Constraint-Set Utility for Partitional Clustering Algorithms. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 115–126. Springer, Heidelberg (2006)
10. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A Density-Based Algortihm for Discovering Clusters in Large Spatial Database with Noise. In: Proc. of KDD (1996)
11. Halkidi, M., Gunopulos, D., Kumar, N., Vazirgiannis, M., Domeniconi, C.: A Framework for Semi-Supervised Learning Based on Subjective and Objective Clustering Criteria. In: Proc. of ICDM (2005)
12. Ruiz, C., Menasalvas, E., Spiliopoulou, M.: Constraint-based Clustering. In: Proc. of AWIC. SCI. Springer (2007)
13. Ruiz, C., Menasalvas, E., Spiliopoulou, M.: C-DenStream: Using Domain Knowledge on a Data Stream. In: Gama, J., Costa, V.S., Jorge, A.M., Brazdil, P.B. (eds.) DS 2009. LNCS, vol. 5808, pp. 287–301. Springer, Heidelberg (2009)
14. Ruiz, C., Spiliopoulou, M., Menasalvas, E.: User Constraints Over Data Streams. In: IWKDDS (2006)
15. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained K-means Clustering with Background Knowledge. In: Proc. of ICML (2001)
16. Zhang, X., Furtlehner, C., Sebag, M.: Data Streaming with Affinity Propagation. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 628–643. Springer, Heidelberg (2008)

Unsupervised Sparse Matrix Co-clustering for Marketing and Sales Intelligence

Anastasios Zouzias[1], Michail Vlachos[2], and Nikolaos M. Freris[2]

[1] Department of Computer Science,
University of Toronto, Canada
[2] IBM Zürich Research Laboratory, Switzerland

Abstract. Business intelligence focuses on the discovery of useful retail patterns by combining both historical and prognostic data. Ultimate goal is the orchestration of more targeted sales and marketing efforts. A frequent analytic task includes the discovery of associations between customers and products. Matrix co-clustering techniques represent a common abstraction for solving this problem. We identify shortcomings of previous approaches, such as the explicit input for the number of co-clusters and the common assumption for existence of a block-diagonal matrix form. We address both of these issues and present techniques for automated matrix co-clustering. We formulate the problem as a recursive bisection on Fiedler vectors in conjunction with an eigengap-driven termination criterion. Our technique does not assume perfect block-diagonal matrix structure after reordering. We explore and identify off-diagonal cluster structures by devising a Gaussian-based density estimator. Finally, we show how to explicitly couple co-clustering with product recommendations, using real-world business intelligence data. The final outcome is a robust co-clustering algorithm that can discover in an automatic manner both disjoint and overlapping cluster structures, even in the preserve of noisy observations.

1 Introduction

Graph structures constitute a prevalent representation form for modeling connections between different entities. In particular, analysis of bi-partite graphs is the focus on a wide spectrum of studies that span from social-network to business analytics and decision making. In business intelligence, bi-partite graphs may capture the connection between sets of customers and sets of products. Analysis of such data holds great importance for companies that collect large amounts of customer interaction data in their data warehouses.

One common analytic process for business intelligence is the identification of groups of customers that buy (or do not buy) a subset of products. The availability of such information is advantageous to both sales and marketing teams, as follows: sales people can use these insights for offering more accurate personalized product suggestions to customers by examining what 'similar' customers buy. In a similar manner, identification of buying/not-buying preferences can

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 591–603, 2012.

assist marketing people to determine groups of customers interested in a set of package products. This can help organize more focused marketing campaigns, hence leading to a more appropriate allocation of the company's marketing funds.

The described problem can be mapped to a co-clustering instance [1,4,11]. A similar task is 'matrix reordering' which discovers a permutation of matrix rows and columns such that the resulting matrix is as 'compact' as possible. We view co-clustering as a matrix reordering with a subsequent clustering step for dense area identification. For this work we provide examples from a particular setting, where the rows represent customers and the columns identify the products that a customer has bought; but our approach is applicable in different settings, too. An example of a matrix reorganization is shown in Figure 1 a) and b). Existence of a 'one' (black dot) signifies that a customer has bought a product, otherwise the value is 'zero' (white dot). It is evident that the reordered matrix view provides strong evidence on the existence of patterns in the data.

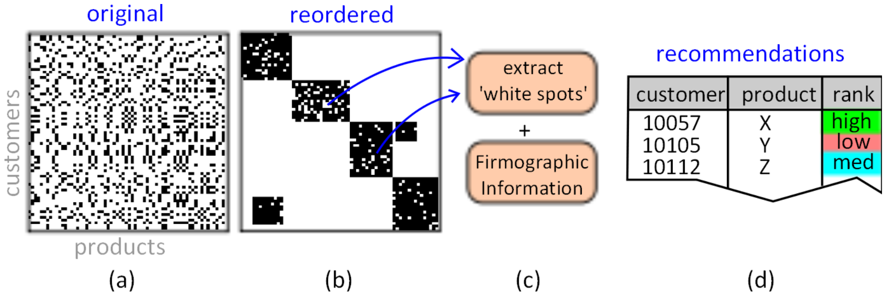


Fig. 1. Overview of our approach. a) Original matrix of customers-products, b) Matrix is reorganized, c) 'White spots' within clusters are extracted and combined with firmographic information, d) Product recommendations are constructed.

Existing co-clustering methods can face several practical issues which limit both their efficiency and the interpretability of the results. For example, the majority of co-clustering algorithms explicitly require as input the number of clusters in the data. In most business scenarios, such an assumption is unrealistic. We cannot assume prior knowledge on the data, but we require a technique that allows data exploration. There exist some methodologies that attempt to perform *automatic* co-clustering [3], i.e., determine the number of co-clusters. They address the problem by evaluating different number of configurations and retaining the solution that provides the best value of the given objective function (e.g., entropy minimization). Such a trial-based approach can significantly affect the performance of the algorithm. Therefore, these techniques are better suited for off-line data processing, rather than for interactive data analysis, which constitutes a key requirement for our setting.

Another shortcoming of many spectral-based approaches is that they typically assume a perfect block-diagonal form for the reordered matrix; "off-diagonal" clusters are usually not detected. This is something that we accommodate in our solution, where existence of possible "off-diagonal" dense clusters is resolved through a Gaussian-based density estimator algorithm. Because we are interested in finding rectangular clusters, the algorithm discovers the parameters of those rectangles (center, width, height) that best cover the highest density of "off-diagonal" areas. Note, that using such an approach we can also support discovery of overlapping co-clusters with no extra effort.

We use the discovered co-clusters for providing product *recommendations* to customers. The customers in our setting are not individuals but large companies, for whom we have extensive information such as company turnover, number of employees, etc. We use this information to prioritize recommendations. Recall that a co-cluster corresponds to a set of customers with similar buying patterns. To this end, existence of 'white spots' within a discovered co-cluster represents potential product recommendations. However, not all white areas within a cluster are equally important. These recommendations need to be *ranked.* We rank the quality and importance of each recommendation based on:

- The quality of the discovered co-cluster. For example, a single white spot in a very dense and large co-cluster is more important than a white spot in a smaller and sparse co-cluster.
- Firmographic and financial characteristics of the customer; recommendations for customers that have bought more products in the past should be ranked higher, because they have exhibited a higher buying propensity.

With the combination of the above two characteristics, the recommendations exploit both *global* patterns as discovered by the co-clustering, and *personalized* metrics.

In summary, our main **contribution** is providing a robust, unsupervised and fast solution for co-clustering which can be used for interactive business intelligence scenarios. We also demonstrate how to use our solution for providing product recommendations. We perform a comprehensive empirical study using real and synthetic data-sets to validate our solutions. To the best of our knowledge, this is one of the few works that evaluate the performance of co-clustering algorithms on real-world, business intelligence data.

2 Related Work

The principle of co-clustering was introduced first by Hartigan with the goal of 'clustering cases and variables simultaneously' [11]. Initial applications were for the analysis of voting data. Hartigan's method is heuristic in nature and may fail to find existing dense co-clusters under certain cases. In [4] the authors present an iterative algorithm that convergences to a local minimum of the same objective function as in [11]. [1] describes an algorithm which provides constant factor approximations to the optimum co-clustering solution using the same objective function. A spectral co-clustering method based on the Fiedler vector appeared in [6]. Our approach uses a similar analytical toolbox as [6], but in addition is *automatic* (number of clusters need not be given), and does not assume perfect block-diagonal form for the matrix. A different approach, views the input matrix as an empirical joint probability distribution of two discrete random variables and poses the co-clustering problem as an optimization problem from an information theoretic perspective [7]. So, the optimal co-clustering maximizes the mutual information between the clustered random variables. A method employing a similar metric as the one of [7] appeared in [20]; the latter approach also returns the co-clusters in a hierarchical format. Finally, [16] provides a parallel implementation of the method of [20] using the map-reduce framework. More detailed reviews on the topic can be found in [14] and [21].

Our approach is equally rigorous with the above approaches; more importantly, it lifts important shortcomings of the spectral-based approaches and in addition focuses on the recommendation aspect of co-clustering, something that previous efforts do not consider.

3 Overview of Our Approach

3.1 Preliminaries

We denote by $\mathbf{I}, \mathbf{0}, \mathbf{1}$ the identity matrix, all-zero, and all-one vector, respectively, and the dimensions will become clear from the context. We define $[m] := \{1, 2, \ldots, m\}$. Let $\boldsymbol{C}$ be an $m \times n$ matrix. For a subset of its rows R and a subset of its columns T we denote by $\boldsymbol{C}_{R,T}$ the sub-matrix formed by rows R and columns T.

Given an undirected graph $G = (V, E)$ on n vertices with adjacency matrix $\boldsymbol{A}$, its Laplacian matrix is defined $\boldsymbol{L} := \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{D}$ is a diagonal matrix of size n with $D_{ii} = \sum_j A_{ij}$. Moreover, the normalized Laplacian matrix of G is defined, when $D_{ii} > 0$ for all i, as $\widehat{\boldsymbol{L}} := \mathbf{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$. The eigenvector of $\widehat{\boldsymbol{L}}$ that corresponds to the second smallest eigenvalue of $\widehat{\boldsymbol{L}}$ is known as the *Fiedler vector* [8]. Let $(S, \bar{S})$ be a bipartition of the vertex set of G. Denote by $\mathrm{cut}(S, \bar{S})$ the sum of weights of the edges between the sets S and $\bar{S}$.

3.2 Graph Partitioning

Partitioning a graph into two balanced vertex sets (i.e., two sets such that neither is much larger than the other one) while minimizing the number of edges between them is a fundamental combinatorial problem with various applications [19]. Choosing a particular *balancing condition* gives rise to different related measures of the quality of the cut including conductance, expansion, normalized and sparsest cut. The two most commonly used balancing objective functions are the *ratio cut* [10] and the *normalized cut* [17]. In ratio cut, the size of a subset $S \subset V$ of a graph is measured by its number of vertices $|S|$, where in normalized cut the size is measured by the total weights of its edges, denoted by $\mathrm{vol}(S)$. Here we will use the normalized cut objective function

$$\mathrm{Ncut}(S, \bar{S}) := \frac{\mathrm{cut}(S, \bar{S})}{\mathrm{vol}(S)} + \frac{\mathrm{cut}(S, \bar{S})}{\mathrm{vol}(\bar{S})}.$$

Note that the above objective function typically takes a small value if the bipartition $(S, \bar{S})$ is not balanced, hence it favors balanced partitions. The goal of graph partitioning is to solve the optimization problem

$$\min_{S \subset V} \mathrm{NCut}(S, \bar{S}). \tag{1}$$

This problem is NP-hard. Many approximation algorithms for this problem have been developed over the last years [2,12], however they turn out not to be performing well in practice. In our approach, we employ a heuristic that is based on spectral techniques and works well in practice [13,9]. Following the notation of [13], for any $S \subset V$, define the vector $\boldsymbol{q} \in \mathbb{R}^n$ as

$$q_i = \begin{cases} +\sqrt{\eta_2/\eta_1}, \, i \in S; \\ -\sqrt{\eta_1/\eta_2}, \, i \in \bar{S}, \end{cases} \tag{2}$$

where $\eta_1 = \mathrm{vol}(S)$ and $\eta_2 = \mathrm{vol}(\bar{S})$. The objective function in (1) can be written (see [6] for details) as follows

$$\min \frac{\boldsymbol{q}^\top \boldsymbol{L} \boldsymbol{q}}{\boldsymbol{q}^\top \boldsymbol{D} \boldsymbol{q}}, \quad \text{subject to } \boldsymbol{q} \text{ as in (2) and } \boldsymbol{q}^\top \boldsymbol{D} \boldsymbol{1} = \boldsymbol{0},$$

where the extra constraint $\boldsymbol{q}^\top \boldsymbol{D} \boldsymbol{1} = \boldsymbol{0}$ excludes the trivial solution where $S = V$ or $S = \emptyset$. Even though the above problem is also NP-hard, we adopt a spectral 2-clustering heuristic: first, we drop the constraint that $\boldsymbol{q}$ is as in Eqn. (2); this relaxation is equivalent to finding the second largest eigenvalue and eigenvector of the generalized eigensystem $\boldsymbol{L}\boldsymbol{z} = \lambda \boldsymbol{D}\boldsymbol{z}$. Then, we round the resulting eigenvector $\boldsymbol{z}$ (a.k.a. *Fiedler vector*) to obtain a bipartition of G. The rounding is performed by applying 2-means clustering separately on the coordinates of $\boldsymbol{z}$, which can be solved *exactly* and efficiently.

4 The Algorithm

The proposed algorithm consists of two steps: in the first step, we compute a permutation of the row set and column set using a recursive spectral bi-partitioning algorithm; in the second step we use the permuted input matrix to identify any remaining clusters by means of a Gaussian-based density estimator.

4.1 Recursive Spectral Bi-partitioning

In this section, we recall a graph theoretic approach to the co-clustering problem [6]. The input is an $m \times n$ matrix $\boldsymbol{C}$. For illustration purposes, we assume that $\boldsymbol{C}$ is a binary matrix, i.e., its elements are in $\{0, 1\}$, but our approach is applicable in general. Given $\boldsymbol{C}$, we can uniquely define a bipartite graph $G = (L \cup R, E)$, $|L| = m$, $|R| = n$ as follows: Each element of the left set of vertices, L, corresponds to a row of $\boldsymbol{C}$ and each element of the right vertices R to a column of $\boldsymbol{C}$. We connect an edge between $i \in L$ and $j \in R$ if and only if $C_{ij} = 1$. Now given the bipartite graph G corresponding to $\boldsymbol{C}$, we find a balanced cut in G with few edges crossing the cut.

Algorithm 1. Recursive Spectral Bipartition

```
1: procedure RecBipart($\boldsymbol{C}$) ▷ $\boldsymbol{C}$: an $m \times n$ binary matrix, $EigenGap$ : a stopping parameter
2:     $\{(R_1, R_2), (T_1, T_2)\}$ = SplitCluster($\boldsymbol{C}$).
3:     if there is a split $(R_1, R_2)$ and $(T_1, T_2)$ then
4:         Run RecBipart($\boldsymbol{C}_{R_1,T_1}$)
5:         Run RecBipart($\boldsymbol{C}_{R_2,T_2}$)
6:     end if
7:     Output: A partition of the row and column set of $\boldsymbol{C}$, i.e., $\cup R_i = [m]$ and $\cup T_i = [n]$.
8: end procedure

9: procedure SplitCluster($\boldsymbol{C}$)                                   ▷ $\boldsymbol{C}$: binary matrix
10:     Let $\widehat{\boldsymbol{L}}$ be the normalized Laplacian of the bipartite graph that corresponds to $\boldsymbol{C}$
11:     Let $\lambda_2$ and $\boldsymbol{z}$ be the second smallest eigenvalue and eigenvector of $\widehat{\boldsymbol{L}}$
12:     if $\lambda_2 > EigenGap$ then
13:         Bipartition the coordinates of $\boldsymbol{z}$ s.t. the sum of its intra-variances is minimized.
14:     end if
15:     Output: A bipartition of the row set and the column set into $(R_1, R_2)$ and $(T_1, T_2)$, respectively.
16: end procedure
```

4.2 Eigengap-Based Termination

A basic fact in spectral graph theory is that the number of connected components in an undirected graph equals to the multiplicity of the zero eigenvalue of its normalized Laplacian matrix. Cheeger's inequality provides an "approximate" version of the latter fact [5]. That is, a graph has a sparse (normalized) cut if and only if there are at least two eigenvalues that are close to zero. Let the conductance of a graph $G = (V, E)$ defined as follows

$$c(G) := \min_{S \subseteq V, \mathrm{vol}(S) \leq \frac{|E|}{2}} \frac{\mathrm{cut}(S, \bar{S})}{\mathrm{vol}(S)}.$$

Cheeger's inequality [5] tells us that $2c(G) \geq \lambda_2 \geq c(G)^2/2$, where λ_2 is the second smallest eigenvalue of the normalized Laplacian of G. The first inequality of the above equation implies that if λ_2 is large, then G does not have sufficiently small conductance. The latter implication supports our choice of the termination criterion of the recursion of Algorithm 1 (see Step 11 of the SplitCluster procedure). Roughly speaking, we want to stop the recursion when the matrix can not be reduced (after permutation of rows and columns) to an approximately block diagonal matrix, equivalently when the bipartite graph associated to the current matrix does not contain any sparse cut. Using Cheeger's inequality, we can efficiently check if the bipartite graph has a sufficiently good cut or not. An illustration of the algorithm's recursion is explored in Fig.2. Our approach has many similarities with Newman's modularity partitioning but it is not identical [15].

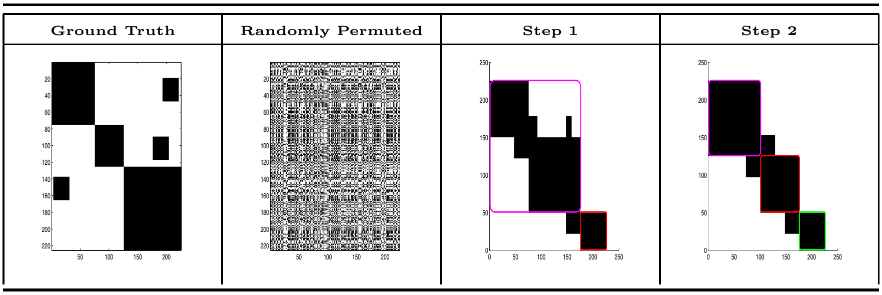


Fig. 2. Running example of Algorithm 1

4.3 Discovering Off-Diagonal Clusters

After termination of Algorithm 1, we expect to have produced a fairly good reordering of the rows and columns of the input matrix $\boldsymbol{C}$ and moreover to have discovered a set of co-clusters that have disjoint row and column sets. However, in almost all instances of practical interest, we cannot expect the set of co-clusters to have disjoint row and column sets; several "off-diagonal" co-clusters may have appeared after the course of our reordering algorithm. In order not to discard this potentially useful information, we apply as a post-processing step a Gaussian-based density estimator on the matrix after removing the set of co-clusters already discovered by Algorithm 1. This process is depicted in Figure 3. In the first step, we remove all the clusters that have been already extracted by Algorithm 1. In the second step we apply a density estimator to discover any (possibly) remaining co-clusters. That is, we convolute a Gaussian mask over all positions of the binary matrix to detect the most dense areas. Initially, we set a sufficiently large size on the Gaussian mask[1]. We then progressively reduce the size of the mask by a fixed constant. At each step, we record all sufficiently dense areas (those for which the convolution exceeds some threshold) and remove the co-cluster from further consideration.

Complexity: We briefly discuss the time complexity of Algorithm 1. For ease of presentation, assume that the input is a square matrix of size n and let $T(n)$ be the time complexity. Moreover, we may assume[2] that in every recursion step the partitioning is balanced. Since the input matrix is sparse, the following recursion holds $T(n) \approx 2T(n/2) + \mathcal{O}(n \log n)$, where the extra additive factor is due to sorting (Lanczos method is used for computing the eigenvector). Solving the recursion we get that $T(n) = \mathcal{O}(n \log^2 n)$ which implies that our method is only more expensive than a single bi-partitioning by a poly-logarithmic factor.

[1] We can over-estimate the size of the largest dense rectangle by the number of non-zero entries of the input matrix.

[2] We are allowed to do so, since a constant number of successive unbalanced cuts indicate that the recursion should terminate.

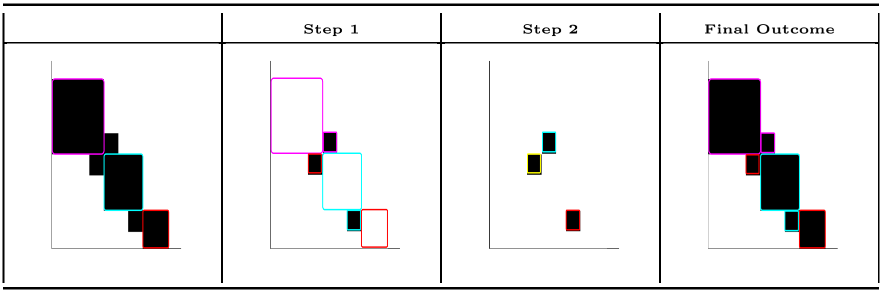


Fig. 3. Discovering Off-diagonal Clusters

4.4 Recommendations

Algorithm 1 together with the density estimation step output a list of co-clusters. In the business scenarios that we consider co-clusters will represent strongly correlated customer and products subsets. We illustrate how this information can be used to drive meaningful product recommendations.

Many discovered co-clusters are expected to contain “white-spots”. These represent customers that exhibit similar buying pattern with a number of other customers, they still have not bought a product within the co-cluster. These are products that constitute good recommendations. Essentially, we exploit the existence of *globally-observable* patterns for making individual recommendations.

Not all “white-spots” are equally important. We rank them by considering firmographic and financial characteristics of the customers. The intuition is that ‘wealthy’ customers/companies that have bought many products in the past are better-suited candidates. They are at financial position to buy a product and they have already established a buying relationship. In our formula we consider three factors:

- **Turnover T** is the revenue of the company as provided in its financial statements.
- **Past Revenue R** is the revenue amount that our company made in its interactions with the customer during the past 3 years.
- **Industry Growth IG** represents that predicted growth for the industry in which the customer belongs for the upcoming year. This data is furnished from marketing databases and is estimated from diverse global financial indicators.

Therefore the rank r of a given white-spot that captures a customer-product recommendation is given by:

$$r = w_1\mathrm{T} + w_2\mathrm{R} + w_3\mathrm{IG}, \quad \sum_i w_i = 1,$$

In our scenario, the weights w_1,w_2,w_3 are assumed to be equal but in general they can be tuned appropriately.

We have described how to rank the "white-spots" within a particular co-cluster. In order to give a total ordering on the set of the recommendations, we should normalize these ranking value with the importance of each co-cluster. We define the importance of each co-cluster as the product of its area and density normalized by the sum of the importance of all co-clusters. Hence, we normalize all recommendations by the importance of the corresponding co-cluster.

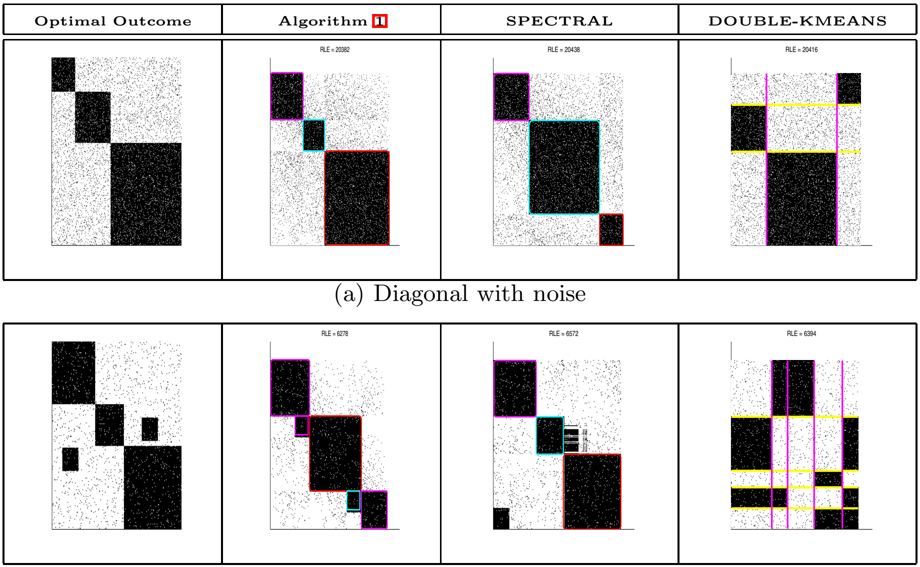
5 Experiments

5.1 Comparison with other Techniques

We compare the proposed approach with two other techniques. The first one (SPECTRAL) is described in [6] and is similar with the proposed approach. The main difference is that the approach of [6] performs k-partition of the input matrix using the eigenvectors that correspond to the smallest eigenvalues. Moreover, in order to compute the clustering it utilizes k-means clustering which makes the approach randomized, compared to our approach which is deterministic. The second one (DOUBLE-KMEANS) is described in [1]. First, it performs k-means clustering using as input vectors the columns of the input matrix and then permutes the columns by grouping together columns that belong in the same cluster. In the second step, it performs the same procedure on the rows of the input matrix using a possible different number of clusters, say l. This approach outputs $k \cdot l$ clusters. Typical values for k and l that we use, are between 3 and 5. We run all the above algorithms on synthetic data which we produced by creating several block-diagonal and off-diagonal clusters. We introduced "salt-and-pepper" noise in the produced matrix, in an effort to examine the accuracy of the compared algorithms even in when diluting the strength of the original patterns. The results are summarized in Figure 4. We observe that our algorithm can detect with high efficiency the original patterns, whereas the original spectral and k-Means algorithms present results of lower quality.

5.2 Compression-Based Evaluation

It is common to judge the effectiveness of a particular algorithm based on the value of an objective function. However, it is not clear how to evaluate the effectiveness of various co-clustering algorithms that are designed to optimize different objective functions. It is even harder to fairly compare the quality of two algorithms that output a different number of co-clusters. Therefore, we make a comparison using compressibility metrics: we measure how many bytes the reordered array will require when stored using Run-Length-Encoding (RLE) [18]. Recall that RLE replaces long blocks of repetitive values with just two numbers: the value and the length of the "run". For example, RLE encodes the sequence $0, 0, 0, 0, 0, 0, 0, 0, 0$ as a tuple $(9, 0)$. This simple metric allows to quantify how appropriately each algorithm packed together zeros and ones. Understandably, larger number of bytes for the reordered matrix under RLE compression, indicates worse performance in placing zeros and ones in adjacent positions.



(a) Diagonal with noise

(b) Diagonal with outliers

Fig. 4. The first column contains the ground truth and the remaining columns contains the output of the three algorithms described in Section 5.1. All algorithms take as input a randomly permuted (independently in rows and columns) version of the ground truth instance.

The results using are depicted in Table 1. For this we extract matrices that represent buying patterns within our company for various industries of customers, because different industries exhibit different patterns. We notice that the proposed recursive algorithm results in compressed matrix sizes significantly smaller than the competitive approaches, suggesting a more effective co-clustering process.

Table 1. We compare the efficacy of the various co-clustering algorithms by reporting the number of bytes when the reordered matrix is compressed using Run-Length-Encoding (RLE). We present results for buying patterns extracted from various customer industries. Our approach results in reordered matrices that can be better compressed.

Industry's name	Original	Our Approach	SPECTRAL	DOUBLE-KMEANS
Computer Services	2004	108	306	544
Professional Services	1136	212	484	658
Banks	2810	372	1012	1348
Provincial Government	954	128	236	352
Other Productions Ind.	1458	288	648	720
Retail	5232	360	888	2148
Travel & Transport	1158	204	534	582
Wholesale distribution services	638	212	408	394

5.3 Business Intelligence on Real Datasets

For this example we use real-world data provided by our sales department relating to approximately 30,000 Swiss customers. The dataset contains all firmographic information pertaining to the customers, such as: industry categorization (electronic, automotive, etc), expected industry growth, customer's turnover for last, past revenue. We perform co-clustering on the customer-product matrix.. We apply our algorithm on each industry separately, because sales people only have access to their industry of specialization. Figure 5 shows the outcome of the algorithm and the detected diagonal and off-diagonal clusters. The highest ranked recommendations are detected within the blue cluster, and they suggest that 'white-spot' customers within this cluster can be approached with an offer for the product 'System-I'. These customers were ranked higher based on their financial characteristics.

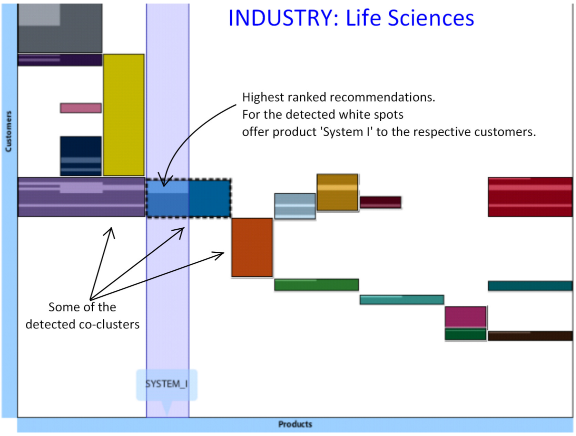


Fig. 5. Example of our co-clustering algorithm when applied on customers of a particular industry (Life Sciences). We see that the algorithm can easily discern off-diagonal clusters. For the illustrated "white-spot" customers within the blue cluster, there is a product recommendation for 'System I'.

6 Conclusion

Focus of this work was to explicitly show how co-clustering techniques can be coupled with recommender systems for business intelligence applications. Contributions of our approach include:

- An unsupervised spectral-based technique for detection of large 'diagonal' co-clusters. We present a robust termination criterion and we depict its accuracy on a variety of synthetic data where we compare with ground-truth.

- A Gaussian-based density estimator for identification of smaller 'off-diagonal' co-clusters.
- A comprehensive comparison of our approach with prevalent co-clustering approaches using a compression-based metric.
- A direct application of our methodology in business recommender systems.

References

1. Anagnostopoulos, A., Dasgupta, A., Kumar, R.: Approximation Algorithms for co-Clustering. In: Proceedings of ACM Symposium on Principles of Database Systems (PODS), pp. 201–210 (2008)
2. Arora, S., Rao, S., Vazirani, U.: Expander Flows, Geometric Embeddings and Graph Partitioning. J. ACM 56, 5:1–5:37 (2009)
3. Chakrabarti, D., Papadimitriou, S., Modha, D.S., Faloutsos, C.: Fully Automatic Cross-associations. In: Proc. of International Conference on Knowledge Discovery and Data Mining (KDD), pp. 79–88 (2004)
4. Cho, H., Dhillon, I.S., Guan, Y., Sra, S.: Minimum Sum-Squared Residue co-Clustering of Gene Expression Data. In: Proc. of SIAM Conference on Data Mining, SDM (2004)
5. Chung, F.R.K.: Spectral Graph Theory. American Mathematical Society (1994)
6. Dhillon, I.S.: Co-Clustering Documents and Words using Bipartite Spectral Graph Partitioning. In: Proc. of International Conference on Knowledge Discovery and Data Mining (KDD), pp. 269–274 (2001)
7. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-Clustering. In: Proc. of International Conference on Knowledge Discovery and Data Mining (KDD), pp. 89–98 (2003)
8. Fiedler, M.: Algebraic Connectivity of Graphs. Czechoslovak Mathematical Journal 23(98), 298–305 (1973)
9. Guattery, S., Miller, G.L.: On the Performance of Spectral Graph Partitioning Methods. In: Proc. of ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 233–242 (1995)
10. Hagen, L., Kahng, A.: New Spectral Methods for Ratio Cut Partitioning and Clustering. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 11(9), 1074–1085 (1992)
11. Hartigan, J.A.: Direct Clustering of a Data Matrix. Journal of the American Statistical Association 67(337), 123–129 (1972)
12. Leighton, T., Rao, S.: Multicommodity Max-flow Min-cut Theorems and their Use in Designing Approximation Algorithms. J. ACM 46, 787–832 (1999)
13. Luxburg, U.: A Tutorial on Spectral Clustering. Statistics and Computing 17, 395–416 (2007)
14. Madeira, S., Oliveira, A.L.: Biclustering Algorithms for Biological Data Analysis: a survey. Trans. on Comp. Biology and Bioinformatics 1(1), 24–45 (2004)
15. Newman, M.E.J.: Fast Algorithm for Detecting Community Structure in Networks. Phys. Rev. E 69, 066133 (2004)
16. Papadimitriou, S., Sun, J.: DisCo: Distributed Co-clustering with Map-Reduce: A Case Study towards Petabyte-Scale End-to-End Mining. In: Proc. of International Conference on Data Mining (ICDM), pp. 512–521 (2008)

17. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
18. Salomon, D.: Data Compression: The Complete Reference, 2nd edn. Springer-Verlag New York, Inc. (2000)
19. Shmoys, D.B.: Cut Problems and their Application to Divide-and-conquer, pp. 192–235. PWS Publishing Co. (1997)
20. Sun, J., Faloutsos, C., Papadimitriou, S., Yu, P.S.: GraphScope: Parameter-free Mining of Large Time-evolving Graphs. In: Proc. of KDD, pp. 687–696 (2007)
21. Tanay, A., Sharan, R., Shamir, R.: Biclustering Algorithms: a survey. Handbook of Computational Molecular Biology (2004)

Expectation-Maximization Collaborative Filtering with Explicit and Implicit Feedback

Bin Wang, Mohammadreza Rahimi, Dequan Zhou, and Xin Wang

Department of Geomatics Engineering, University of Calgary
2500 University Drive NW, Calgary, AB, Canada, T2N 1N4
{bw.wang,smrahimi,dzho,xcwang}@ucalgary.ca

Abstract. Collaborative Filtering (CF) is a popular strategy for recommender systems, which infers users' preferences typically using either explicit feedback (e.g., ratings) or implicit feedback (e.g., clicks). Explicit feedback is more accurate, but the quantity is not sufficient; whereas implicit feedback has an abundant quantity, but can be fairly inaccurate. In this paper, we propose a novel method, *Expectation-Maximization Collaborative Filtering* (EMCF), based on matrix factorization. The contributions of this paper include: first, we combine explicit and implicit feedback together in EMCF to infer users' preferences by learning latent factor vectors from matrix factorization; second, we observe four different cases of implicit feedback in terms of the distribution of latent factor vectors, and then propose different methods to estimate implicit feedback for different cases in EMCF; third, we develop an algorithm for EMCF to iteratively propagate the estimations of implicit feedback and update the latent factor vectors in order to fully utilize implicit feedback. We designed experiments to compare EMCF with other CF methods. The experimental results show that EMCF outperforms other methods by combining explicit and implicit feedback.

1 Introduction

In the modern digital world, consumers are overwhelmed by the huge amount of product choices offered by electronic retailers and content providers. Recommender systems, which analyze patterns of user interests in products in order to provide personalized recommendatons satisfying users' tastes, have recently attracted a great deal of attention from both academia and industry. *Collaborative filtering* (CF) [9], which analyzes relationships among users and items (i.e., products) in order to identify potential associations between users and items, is a popular strategy for recommender systems. Compared to *content filtering* [8], which is the other recommendation strategy that depends on profiles of users and/or items, CF has the advantage of being free of domain knowledge. Since CF relies only on the history of user behavior, it can address the issues in creating explicit profiles, which are difficult in many recommender system scenarios.

The history of user behavior for CF usually consists of user feedback, which generally refers to any form of user action on items that may convey the information about users' preferences of items. There are two kinds of user feedback:

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 604–616, 2012.

explicit feedback and *implicit feedback*. Explicit feedback is often in the form of rating actions. For example, Amazon.com asks users to rate their purchased CDs and books on a scale of 1-5 stars. Since explicit feedback directly represents users' judgments on items in a granular way, it has been widely used in many traditional CF recommender systems [2] [9] [10]. However, explicit feedback requires users to perform extra rating actions, which may lead to inconvenience for the user. Given the overwhelming amount of items, it is burdensome for users to rate every item they like or dislike.

Implicit feedback, on the other hand, does not need additional rating actions. Implicit feedback generally refers to any user behavior that indirectly expresses user interests. For example, a news website may cache a user's clicking records when browsing news articles, in order to predict what kind of news the user may prefer. Other forms of implicit feedback include keyword searching, mouse movement, and even eye tracking. Since it is relatively easier to collect such user behaviors, implicit feedback attracts the interest of researchers who attempt to infer user preferences from the much larger amount of implicit feedback. However, implicit feedback is less accurate than explicit feedback. For example, a user may regret buying a product online after receiving the real product. It is difficult to determine whether a user likes a product only based on the purchase behavior, even though the user paid for the product.

Explicit feedback and implicit feedback are naturally complementary to each other. With explicit feedback, the quality is more reliable, but the quantity is limited. However, the quality of implicit feedback is less accurate, but there is an abundant quantity. Most of the existing works have solely considered either explicit feedback [3] or implicit feedback [4] [7] in recommender systems. While there are few works that have tried to unify explicit and implicit feedback, explicit feedback is just treated as a special kind of implicit feedback; and, the implicit feedback is simply normalized to a set of numeric rating values. Without carefully studying how to organically combine explicit and implicit feedback, we will not be able to further improve the performances of recommender systems.

In this paper, we propose a novel recommender method based on matrix factorization [5], called expectation-maximization collaborative filtering (EMCF). The first contribution of this paper is that we combine explicit and implicit feedback together, in which both explicit feedback and implicit feedback are fully utilized. The second contribution is that we observe different cases of implicit feedback and develop the corresponding solutions to estimate implicit feedback for different cases. The third contribution is that we design an expectation-maximization-styled algorithm in EMCF to update the estimations of implicit feedback and latent factor vectors.

Instead of treating explicit feedback as special implicit feedback, EMCF initializes a latent factor model with explicit feedback and then updates the latent factor vectors based on the explicit feedback ratings and the implicit feedback estimations. The key challenge in utilizing implicit feedback in matrix factorization is that implicit feedback does not have the numeric rating value. Instead of simply normalizing implicit feedback to a set of numeric rating values, EMCF estimates implicit feedback rating values based on the explicit feedback and available latent factor vectors.

We observe that the implicit feedback can be categorized into four cases, in terms of the distribution of latent factor vectors from the currently trained latent factor model. For different cases, EMCF has corresponding solutions, which are not only based on explicit feedback ratings and implicit feedback estimations, but are also based on the graph-based structure of explicit and implicit feedback.

We also observe that only part of implicit feedback ratings can be estimated based on the current situation of the model. Therefore, an expectation-maximization-styled algorithm is designed in EMCF to: 1) propagate current estimations of implicit feedback plus explicit feedback ratings towards the set of implicit feedback that have not yet been estimated, so that more implicit feedback estimations can be added into the model training; 2) Re-train the latent factor model based on all the available implicit feedback estimations and explicit feedback ratings and then update the latent factor vectors of users and items for further estimating. The algorithm not only fully utilizes implicit feedback with explicit feedback, but also prevents noisy implicit feedback from affecting the performance of the EMCF model.

Experiments have been conducted to compare the EMCF model with other popular models. The experimental results show that EMCF outperforms those models, especially when the percentage of explicit feedback is small.

The rest of the paper is organized as follows: in Section 2, a preliminary is given including the formalization, the background of CF, and a related method called co-rating; in Section 3, we present the observations of implicit feedback and propose the solutions to estimate implicit feedback for different cases; in Section 4, we describe the EMCF model and introduce the algorithm to train the model; in Section 5, the experiments make the comparisons between EMCF and other models; the conclusion and some future works are given in Section 6.

2 Preliminary

2.1 Formalization

In this paper, we use $U = \{u_1, u_2, \dots, u_m\}$ to denote a set of m users and use $I = \{i_1, i_2, \dots, i_n\}$ to denote a set of n items. The explicit feedback and implicit feedback are defined as the observable actions from U to I that may directly and indirectly reflect users' preferences of the items.

Explicit feedback is usually in the form of rating actions. A user rates items by assigning numeric rating values. The observed rating values are represented by a matrix $R \in \Re^{m \times n}$, in which each entry $r_{ui} \in \Re$ is used to denote the rating on item i given by user u. We use S^E to denote the set of explicit feedback, which consists of user-item-rating triples (u, i, r_{ui}).

Implicit feedback typically consists of various types of actions performed by users on items that can be automatically tracked by systems. In some related works [4] [6], implicit feedback is represented by a binary variable $b_{ui} \in \{0, 1\}$, in which 1 means user u performed some action on item i and 0 means u never touched i. In this paper, we assume that a user has no interest to an item if he/she never touched this item. We use S^I to denote the set of implicit feedback, which consists of user-item pairs (u, i) for which u has implicit feedback on i.

2.2 Collaborative Filtering

There are two primary types of CF methods: *nearest-neighbor methods* and *matrix factorization methods*. A brief introduction of these two types of methods is given in this section.

Nearest-neighbor Methods. Nearest-neighbor methods are prevalent in CF [2]. Generally, the procedures of nearest-neighbor methods follow a similar pattern: first calculate the similarity, which reflects distance, correlation, or weight, between two users or two items; then produce a prediction for the active user by taking the weighted average of all the ratings.

In terms of different focuses in the similarity calculation, there are two types of nearest-neighbor methods: *item-based methods* and *user-based methods*. As the name suggests, item-based methods calculate the similarities between items, try to find nearest-neighbor items for the target item, and then evaluate the active user's rating on the target item based on the ratings of its neighbors. Similarly, user-based methods first identify nearest-neighbor users for the target user by calculating the similarities between users, and then make predictions for the target user to unrated items based on neighbor users' ratings.

Matrix Factorization Methods. Matrix factorization is the other primary type of approaches for CF. Latent factor models, which try to explain the rating generation by vectors of latent factors inferred from the patterns of ratings, are typically used in matrix factorization methods. In a sense, such latent factors correspond to the dimensions in a latent space in which the profiles of both users and items can be characterized. Koren et al. [5] gave some examples to interpret latent factors: if the items are movies, the latent factors may measure obvious dimensions such as comedy versus drama, less well-defined dimensions such as depth of character development or quirkiness, or completely un-interpretable dimensions; for users, each latent factor may measure how a user scores the corresponding movie factor.

Matrix factorization methods map both users and items to a joint latent factor space, so that each user is modeled by a user latent factor vector and each item is modeled by an item latent factor vector. The rating for a user-item pair is modeled as an inner product of this user's latent factor vector and this item's latent factor vector.

2.3 Co-rating

Liu et al. [6] developed a matrix factorization model called *co-rating*, which tries to unify explicit and implicit feedback. Co-rating treats explicit feedback as a special kind of implicit feedback, so that the entire set of explicit and implicit feedback can be used simultaneously during the model training. For solving the challenge of implicit feedback ratings having only binary values instead of numeric values, co-rating normalizes the rating values of explicit feedback and the binary values of implicit feedback into a range of $[0, 1]$. With the co-rating method, latent factor vectors are learned by solving an objective function in the matrix factorization model trained with explicit and implicit feedback. The co-rating's objective function is different from the one normally used in other matrix factorization methods [1] [5]; an extra weighted term has been added, which aims at controlling the loss when treating explicit feedback as implicit feedback.

3 Explicit and Implicit Feedback

3.1 Matrix Factorization with Explicit Feedback

In this paper, we fully utilize explicit feedback and implicit feedback together to train a latent factor model by the matrix factorization method. In the matrix factorization method, each item i is associated with a latent factor vector $q_i \in \Re^k$, and each user u is associated with a latent factor vector $p_u \in \Re^k$, where k is the number of latent factors. For a given item i, the elements of q_i measure the extent to which the item possesses those factors. For a given user u, the elements of p_u measure the extent of u's interest in items according to the corresponding factors. The rating value r_{ui} is approximated by the dot product $q_i^{\intercal} p_u$. Therefore, the rating matrix R is approximated by the product of the user latent factor matrix $M_U \in \Re^{k \times m}$ and the item latent factor matrix $M_I \in \Re^{k \times n}$ as

$$R \approx M_U^{\intercal} \cdot M_I. \tag{1}$$

The matrix factorization method learns the individual latent factor vectors in M_U and M_I by solving

$$argmin_{q^*,p^*} \sum_{(u,i,r_{ui}) \in S^T} (r_{ui} - q_i^{\intercal} p_u) + \lambda(\|q_i\|^2 + \|p_u\|^2), \tag{2}$$

where S^T is the training set including the known rating values, the term $r_{ui} - q_i^{\intercal} p_u$ is the estimation of the goodness of the rating approximation, $\|q_i\|^2 + \|p_u\|^2$ is the regularization term to avoid model overfitting, and the constant λ is to control the extent of regularization.

Before utilizing implicit feedback, we first use the set of explicit feedback S^E to initially train the model, so that $S^T \leftarrow S^E$ at this stage. There are two approaches to solve Equation 2: *Stochastic Gradient Descent* (SGD) [5] and *Alternating Least Squares* (ALS) [11].

SGD is a popular approach that is easy to implement and has a relatively fast running time. In SGD, the algorithm loops through all ratings in S^T. For each triple (u, i, r_{ui}), the algorithm computes the associated prediction error:

$$e_{ui} = r_{ui} - q_i^\intercal p_u. \tag{3}$$

The algorithm then modifies the parameters by a magnitude proportional to γ in the opposite direction of the gradient as:

$$q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i) \tag{4}$$

$$p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u) \tag{5}$$

ALS is a different style of algorithm for learning the latent factor vectors. In ALS, one of the unknown latent factor vectors is fixed in order to learn the other vector; and, the latter vector is then fixed to learn the former vector. The procedure is repeated until convergence is reached. With ALS, the optimization of Equation 2 becomes quadratic and can be optimally solved. Although ALS is slower and more complicated than SGD, it is usually favorable when parallelization is needed. Due the space limitation, we are not going to give the details of ALS here.

3.2 Enhance Explicit Feedback with Implicit Feedback

Matrix factorization methods require numeric rating values to learn the latent factor vectors. Although explicit feedback satisfies this requirement, the amount of explicit feedback ratings is usually not sufficient to train an accurate model. On the other hand, the amount of implicit feedback is much larger than the amount of explicit feedback due to the lack of additional rating actions. If there is a way to assign a meaningful numeric estimation to implicit feedback, it can be used to enhance the model trained only based on explicit feedback ratings.

After initializing the model based on explicit feedback ratings, latent factor vectors can be attained for users and items that are included in S^E. For each user-item pair (u, i) in S^I, there are four possible cases:

- **Case 1**: Both u and i have been assigned latent factor vectors p_u and q_i, respectively, because u and i are also included in S^E.
- **Case 2**: u has been assigned latent factor vector p_u because u is also included in S^E, but i has no latent factor vector since i is not included in S^E.
- **Case 3**: i has been assigned latent factor vector q_i because i is also included in S^E, but u has no latent factor vector since u is not included in S^E.
- **Case 4**: Both u and i have no latent factor vectors, because u and i are not included in S^E.

These four cases are demonstrated in Figure 1.

For Case 1, the target implicit feedback can be straightforwardly estimated using the latent factor vectors of u and i. The estimation $\hat{r}^I_{ui}$ can be computed as:

$$\hat{r}^I_{ui} = q_i^\intercal p_u. \tag{6}$$

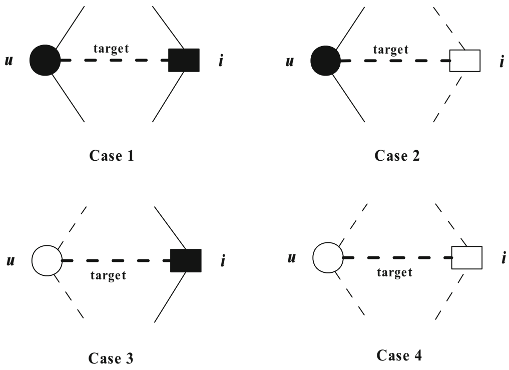


Fig. 1. Four cases of implicit feedback. The black circles and black squares are used to represent users and items, respectively, on which the latent factor vectors have been assigned, and the white circles and white squares are used to represent the users and items, on which there is no latent factor vector assigned yet. The solid lines represent the explicit feedback, and the dash lines represent the implicit feedback. The thick dash lines are the targets of implicit feedback that we will estimate based on the current situation.

For Case 2, the target implicit feedback cannot be directly estimated using latent factor vectors due to the lack of q_i. We use an item-based CF method to estimate it. First, the similarity $sim(i, j)$ between item i and item j is calculated using Jaccard Similarity Coefficient as:

$$sim(i,j) = \frac{|A_i \bigcap A_j|}{|A_i \bigcup A_j|}, \tag{7}$$

where A_i and A_j are the set of users who have either explicit or implicit feedback actions on i and j respectively. Next, we look for the set of neighbor items N_i of item i. In N_i, each neighbor item j has to satisfy the conditions as: 1) the similarity $sim(i, j)$ is larger than a pre-defined threshold; 2) a latent factor vector has already been assigned on j. Then, the estimation $\hat{r}^I_{ui}$ for the target implicit feedback can be computed as:

$$\hat{r}^I_{ui} = \frac{\sum_{j \in N_i} sim(i,j) q_j^\intercal p_u}{\sum_{j \in N_i} sim(i,j)}. \tag{8}$$

Similarly for Case 3, the similarity $sim(u, v)$ between user u and user v is also calculated by Jaccard Similarity Coefficient as:

$$sim(u,v) = \frac{|A_u \bigcap A_v|}{|A_u \bigcup A_v|}, \tag{9}$$

where A_u and A_v are the set of items on which u and v have either explicit or implicit feedback actions respectively. The set of neighbor users N_u for user

u is found, in which each neighbor user v has a value $sim(u,v)$ larger than a threshold and has an assigned latent factor vector. The estimation $\hat{r}^I_{ui}$ for the target implicit feedback can be computed using a user-based CF method as

$$\hat{r}^I_{ui} = \frac{\sum_{v \in N_u} sim(u,v) q_i^\intercal p_v}{\sum_{v \in N_u} sim(u,v)}. \tag{10}$$

For Case 4, there is no sufficient information to estimate the target implicit feedback based on the current situation.

4 Expectation-Maximization Collaborative Filtering

The user neighbor set N_u and the item neighbor set N_i may have no eligible members. Although some user or item is similar enough with the target user or item, they still can not become eligible neighbors due to lack of latent factor vectors. On the other hand, estimations from Equation 8 and 10 are based on currently learned latent factor vectors, and latent factor vectors need to be updated based on the updated training set, in which new estimations will be added in. To address these issues, we design the *Expectation-maximization Collaborative Filtering* (EMCF) algorithm. The basic idea is to iteratively propagate the available implicit feedback estimations, plus the explicit feedback, towards the unavailable implicit feedback, in order to make possible the estimations on such implicit feedback.

If we treat CF model as the objective and treat latent factor matrices as estimated parameters, we can map the classic EM into our problem scenario. Our goal is to build CF model that uses the matrix factorization method. The model depends on both user latent factor vectors and item latent factor vectors. The parameters that we use to estimate latent factor vectors are explicit feedback ratings and implicit feedback estimations. The two steps are defined as the following:

- **E Step:** Train the collaborative filtering model using all the explicit feedback ratings and currently available estimations of implicit feedback.
- **M Step:** Estimate the implicit feedback based on latent factor vectors that are output from the CF model trained in the previous E Step.

The EMCF algorithm is an iterative procedure. We begin by using explicit feedback ratings to initialize the CF model with the matrix factorization method, which is introduced in Section 3.2. The set of implicit feedback is then categorized based on four cases (Cases 1-4 above), in terms of whether the involved users and item have latent factor vectors or not. Following the estimation methods introduced in Section 3.3, the implicit feedback ratings are estimated if they are eligible. The estimated implicit feedback ratings are put together with explicit feedback ratings to train the EMCF model again. Thus, for the user and/or the item involved in the target implicit feedback, new latent factor vectors are assigned if the user and/or the item do not yet have them. For other

```
Algorithm EMCF
Input: Explicit feedback set $S^E$, implicit feedback set $S^I$, user set $U$, and item set $I$.

Output: User latent factor matrix $M^U$ and item latent factor matrix $M^I$.

Initialization:

  – Initialize training set $S^T$ with $S^E$, train the latent factor vectors for users and items in $S^T$, and assign them
    back to $M^U$ and $M^I$.
  – Initialize an empty set $\hat{S}^E$, in which the implicit feedback estimation triples $(u, i, \hat{r}_{ui}$ will be included.

BEGIN
Repeat:
    For each user-item pair $(u, i)$ in $S^I$:
        If both $u$ and $i$ have latent factor vectors:
            Estimate rating $\hat{r}_{ui}$ for $(u, i)$;
            Put $(u, i, \hat{r}_{ui})$ in $\hat{S}^E$ by Equation 6;
            Remove $(u, i)$ from $S^I$;
        Else If $u$ has latent factor vector but $i$ not:
            Estimate rating $\hat{r}_{ui}$ for $(u, i)$ by Equation 8 when item neighbors of $i$ can be found;
            Put $(u, i, \hat{r}_{ui})$ in $\hat{S}^E$;
            Remove $(u, i)$ from $S^I$;
        Else If $i$ has latent factor vector but $u$ not:
            Estimate rating $\hat{r}_{ui}$ for $(u, i)$ by Equation 10 when user neighbors of $u$ can be found;
            Put $(u, i, \hat{r}_{ui})$ in $\hat{S}^E$;
            Remove $(u, i)$ from $S^I$;
    Train model using $S^T \leftarrow S^T \bigcup \hat{S}^E$;
    Update the corresponding columns of $M^U$ and $M^I$ by updated latent factor vectors;
    Evaluate the difference between the rating estimations produced by previous latent factor vectors and the
    rating estimations produced by current latent factor vectors;
Until there is no new entry added in $\hat{S}^E$ and the estimation difference is lower than the threshold.
END
```

Fig. 2. The formal description of EMCF algorithm

users and items that already have latent factor vectors, their latent factor vectors are updated, since the EMCF model is re-trained using the updated rating set. Therefore, the estimations of some non-eligible implicit feedback in the previous round become possible. Then, EMCF algorithm is back to the step of estimating implicit feedback, and the above steps are repeated. The algorithm is terminated when there is no longer eligible implicit feedback to estimate and the rating estimation difference between the previous round and current round is lower than a pre-defined threshold. The formal algorithm procedure description is shown in Figure 2.

The EMCF algorithm has advantages by combining explicit feedback with implicit feedback. First, the implicit feedback is categorized into the four disjoint sets, in terms of the current situations of user and item latent factor vectors. Therefore, we have a chance to deal with implicit feedback differentially. Second, the estimation methods for different cases not only depend on the rating calculation from the matrix factorization, but also consider the neighbor structure built by both explicit feedback and implicit feedback. Third, the iterative procedure of EMCF fully utilizes implicit feedback by providing the opportunity to include more estimations of implicit feedback, which are not eligible in the previous operational round of the algorithm. Finally, the EMCF algorithm prevents noisy implicit feedback from the model training procedure, so that the performance of output model can be improved. Some implicit feedback is not used, since there is no sufficent information for estimation. Usually, such implicit feedback is suspected of being noise.

5 Experiments

We design experiments to demostrate the performance of EMCF. MovieLens[1] is used in the experiments. The dataset consists of one million ratings from 6,000 users and 4,000 movies. In the dataset, 20% ratings are randomly selected and held as the testing set, and the other 80% ratings are used as the training set. In the training set, we follow the idea proposed in [1] to create implicit feedback from explicit ratings data by considering whether a movie is rated by a user. In each experiment, the percentage of implicit feedback is pre-defined, and the rest of ratings in the training set are used as explicit feedback. The experiments are conducted on Apache Mahout[2], which is a recently popular machine learning platform. On Mahout, we implement matrix factorization and co-rating using ALS, and implement EMCF using SGD. All the methods are trained based on the same sets of explicit and implicit feedback, and are tested on the same sets of ratings. The root mean square error (RMSE) has been used as the evaluation measure to compare the methods.

The first set of experiments aims at comparing the performances of EMCF when only considering individual cases of implicit feedback. Given 20% of training set as explicit feedback, the baseline is the matrix factorization only using explicit feedback, which is used to compare to EMCF with different cases of implicit feedback. The results are shown in Table 1.

Table 1. Given 20% explicit feedback, experimental results of matrix factorization only with explicit feedback (MF+Explicit), EMCF with implicit feedback of Case 1, EMCF with implicit feedback with Case 2, EMCF with implicit feedback of Case 3, EMCF with implicit feedback with Case 2 and Case 3, and EMCF with all the implicit feedback

	MF + Explicit	EMCF + Case1	EMCF + Case2	EMCF + Case3	EMCF + Cases2+3	EMCF + All
RMSE	1.039	1.048	0.985	0.990	0.968	0.945

From the results, we can see that the performance of EMCF with implicit feedback of Case 1 is worse than the baseline. It is because the implicit feedback of Case 1 is estimated by the latent factor vectors learned from the model only based on explicit feedback. Without estimations of other cases of implicit feedback, EMCF with Case 1 overfits the model. EMCF with implicit feedback of Case 2 or Case 3 outperforms the baseline. But the improvements of performance are not obvious. EMCF with the implicit feedback combination of Case 2 and Case 3 has a greater improvements compared to the baseline. However, the implicit feedback is not fully utilized due to the lack of Case 1. EMCF with all the implicit feedback has the best performance since the EM-style algorithm of EMCF fully utilizes all the implicit feedback.

[1] http://www.grouplens.org/node/73

[2] http://mahout.apache.org/

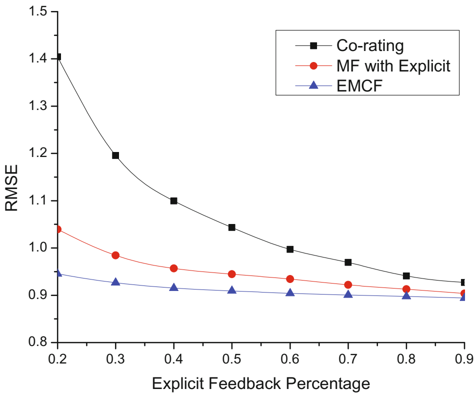


Fig. 3. The experimental results of matrix factorization (MF) only with explicit feedback, co-rating, and EMCF with different percentage of explicit feedback

The second set of experiments aims at comparing the performances of EMCF with different percentages of explicit feedback. There are two baseline methods: matrix factorization only with explicit feedback (MF with Explicit) and co-rating [6]. There are several differences between the co-rating method and our EMCF method. First, EMCF does not treat explicit feedback as a form of special implicit feedback. Instead, EMCF initializes a latent factor model with explicit feedback and then updates the latent factor vectors based on explicit feedback ratings and implicit feedback estimations. Second, EMCF does not normalize explicit feedback and implicit feedback into the same scale. On the contrary, EMCF estimates the ratings of implicit feedback in terms of the scale of explicit feedback. Third, EMCF does not add any new term in the objective function of the matrix factorization. The proposed Expectation-maximization-styled algorithm in EMCF ensures that the estimations of implicit feedback can be adjusted in order to improve the performance of EMCF. The experimental results are shown in Figure 3.

From the results, we can see that EMCF outperforms the other two baseline methods, especially when the percentage of implicit feedback is small. On one hand, the results of the comparison between EMCF and matrix factorization show that utilizing implicit feedback with explicit feedback truly outperforms the method based only on explicit feedback. On the other hand, the results of the comparison between EMCF and co-rating show that simply treating explicit feedback as implicit feedback hurts the performance of the method. The reason may be due to too much noisy implicit feedback added into the model training. The EMCF not only differentially estimates implicit feedback, but also iteratively updates the estimations based on both explicit and implicit feedback, by which noisy implicit feedback can be prevented from affecting the performance.

6 Conclusion and Future Works

In this paper, we present a novel method, Expectation-Maximization Collaborative Filtering (EMCF), which combines explicit and implicit feedback using matrix factorization for recommender systems. EMCF is based on the fact that explicit feedback and implicit feedback are naturally complementary to each other, since explicit feedback has good accuracy, but the quantity is insufficient; whereas, implicit feedback has abundant quantity, but does not have good accuracy. After initializing the EMCF model with explicit feedback, we observe that the implicit feedback can be categorized into four cases, in terms of the distribution of latent factor vectors. We propose three methods to differentially estimate the implicit feedback for the different cases. An EM-styled algorithm is then designed to iteratively propagate the implicit feedback estimations and update the latent factor vectors based on all the available explicit feedback ratings and implicit feedback estimations. We conduct experiments to compare EMCF with two other baseline methods. The experimental results show that EMCF outperforms the other two baselines, especially when the explicit feedback percentage is small.

In the future, we will continue to study how explicit feedback should be combined with implicit feedback in recommender systems. We will adapt different recommender methods for explicit and implicit feedback. We will also consider more complicated types of implicit feedback, such as mouse movements, and more features of implicit feedback, such as the durations of implicit feedback actions.

References

1. Bell, R.M., Koren, Y.: Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights, Los Alamitos, CA, USA, pp. 43–52 (2007)
2. Desrosiers, C., Karypis, G.: A Comprehensive Survey of Neighborhood-based Recommendation Methods. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 107–144. Springer, Boston (2011)
3. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Transactions on Information Systems (TOIS) 22, 89–115 (2004)
4. Hu, Y., Koren, Y., Volinsky, C.: Collaborative Filtering for Implicit Feedback Datasets. In: IEEE International Conference on, Los Alamitos, CA, USA, pp. 263–272 (2008)
5. Koren, Y., Bell, R., Volinsky, C.: Matrix Factorization Techniques for Recommender Systems. Computer 42(8), 30–37 (2009)
6. Liu, N.N., Xiang, E.W., Zhao, M., Yang, Q.: Unifying explicit and implicit feedback for collaborative filtering, New York, NY, USA, pp. 1445–1448 (2010)
7. Pan, R., Scholz, M.: Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD, Paris, France, p. 667 (2009)

8. Pazzani, M.J., 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)
9. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. In: Advances in Artificial Intelligence (2009)
10. Wang, J., Robertson, S., Vries, A.P., Reinders, M.J.T.: Probabilistic relevance ranking for collaborative filtering. Information Retrieval 11(6), 477–497 (2008)
11. Zhou, Y., Wilkinson, D., Schreiber, R., Pan, R.: Large-Scale Parallel Collaborative Filtering for the Netflix Prize. In: Fleischer, R., Xu, J. (eds.) AAIM 2008. LNCS, vol. 5034, pp. 337–348. Springer, Heidelberg (2008)

Author Index