Wen-Chih Peng · Haixun Wang
James Bailey · Vincent S. Tseng
Tu Bao Ho · Zhi-Hua Zhou
Arbee L.P. Chen (Eds.)

LNAI 8643

Trends and Applications in Knowledge Discovery and Data Mining

PAKDD 2014 International Workshops: DANTH, BDM, MobiSocial, BigEC, CloudSD, MSMV-MBI, SDA, DMDA-Health, ALSIP, SocNet, DMBIH, BigPMA
Tainan, Taiwan, May 13–16, 2014
Revised Selected Papers

Springer

More information about this series at http://www.springer.com/series/1244

Wen-Chih Peng · Haixun Wang
James Bailey · Vincent S. Tseng
Tu Bao Ho · Zhi-Hua Zhou
Arbee L.P. Chen (Eds.)

Trends and Applications in Knowledge Discovery and Data Mining

PAKDD 2014 International Workshops: DANTH, BDM, MobiSocial, BigEC, CloudSD, MSMV-MBI, SDA, DMDA-Health, ALSIP, SocNet, DMBIH, BigPMA Tainan, Taiwan, May 13–16, 2014 Revised Selected Papers

Springer

Editors
Wen-Chih Peng
National Chiao Tung University
Hsinchu
Taiwan

Haixun Wang
Google Research
Mountain View, CA
USA

James Bailey
University of Melbourne
Melbourne, VIC
Australia

Vincent S. Tseng
National Cheng Kung University
Tainan
Taiwan

Tu Bao Ho
Japan Advanced Institute of Science
and Technology
Nomi City
Japan

Zhi-Hua Zhou
Nanjing University
Nanjing
China

Arbee L.P. Chen
National Chengchi University
Taipei
Taiwan

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-3-319-13185-6 ISBN 978-3-319-13186-3 (eBook)
DOI 10.1007/978-3-319-13186-3

Library of Congress Control Number: 2014956860

LNCS Sublibrary: SL7 – Artificial Intelligence

Springer Cham Heidelberg New York Dordrecht London

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media
(www.springer.com)

Preface

This volume contains papers presented at PAKDD Workshops 2014 in conjunction with the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) held on May 13, 2014 in Tainan, Taiwan. PAKDD has established itself as the premier event for data mining researchers in the Pacific-Asia region. PAKDD 2014 has 13 workshops and these workshops are Data Analytics for Targeted Healthcare (DANTH), Data Mining and Decision Analytics for Public Health and Wellness (DMDA-Health), Biologically Inspired Data Mining Techniques (BDM), Mobile Data Management, Mining, and Computing on Social Networks (MobiSocial), Big Data Science and Engineering on E-Commerce (BigEC), Cloud Service Discovery (CloudSD), Mobile Sensing, Mining and Visualization for Human Behavior Inferences (MSMV-HBI), Scalable Dats Analytics: Theory and Algorithms (SDA), Algorithms for Large-Scale Information Processing in Knowledge Discovery (ALSIP), Data Mining in Social Networks (SocNet), Data Mining in Biomedical informatics and Healthcare (DMBIH), Pattern Mining and Application of Big Data (BigPMA), and Pacific Asia Workshop on Intelligence and Security Informatics (PAISI). This volume collected the revised papers from the first 12 workshops. The papers of PAISI are included in a separate proceeding.

The total number of submissions for PAKDD 2014 workshops was 179. All papers were reviewed by at least two reviewers. Among 179 paper submissions, only 73 papers were accepted for presentation, and their revised versions are collected in this volume. The acceptance rate was approximately 40.78 %. The general quality of submissions was high and the competition was tough. The workshops would not be successful without the support of authors, reviewers, and organizers. We thank the many authors for submitting their research papers to the PAKDD workshops. We thank the successful authors whose papers are published in this volume for their collaboration in paper revision. We appreciate all Program Committee members for their timely reviews working to a tight schedule. We also thank members of the organization committees for organizing paper submission, reviewing, discussion, feedback, and the final submission. We appreciate the professional service provided by the Springer LNCS editorial and publishing teams, and Miss Anna Kramer's assistance.

June 2014

Wen-Chih Peng
Haixun Wang
James Bailey
Vincent S. Tseng
Tu Bao Ho
Zhi-Hua Zhou
Arbee L.P. Chen

Organization

PAKDD Conference Chairs

Zhi-Hua Zhou	Nanjing University, China
Arbee L.P. Chen	National Chengchi University, Taiwan

Workshop Chairs

Wen-Chih Peng	National Chiao Tung University, Taiwan
Haixun Wang	Google Inc., USA
James Bailey	University of Melbourne, Australia

DANTH Chairs

Osmar Zaïane	University of Alberta, Canada
Dajun (Daniel) Zeng	Chinese Academy of Sciences, China and University of Arizona, USA
Ji Zhang	University of Southern Queensland, Australia
Guandong Xu	University of Technology, Sydney, Australia
Xiaohui Tao	University of Southern Queensland, Australia
Yidong Li	Beijing Jiaotong University, China

BDM Chairs

Nik Kasabov	AUT University, New Zealand
Shafiq Alam Burki	University of Auckland, New Zealand
Gillian Dobbie	University of Auckland, New Zealand
Yun Sing Koh	University of Auckland, New Zealand

MobiSocial Chairs

De-Nian Yang	Academia Sinica, Taiwan
Wang-Chien Lee	Pennsylvania State University, USA

BigEC Chairs

Chih-Chieh Hung	Rakuten Inc., Japan
Jie Tang	Tsinghua University, China
Yi Chang	Yahoo! Labs, USA

CloudSD Chairs

Jian Wu	Zhejiang University, China
Zibin Zheng	The Chinese University of Hong Kong, Hong Kong
Liang Chen	Zhejiang University, China
Qi Yu	Rochester Institute of Technology, USA

MSMV-MBI Chairs

Edward Y. Chang	HTC Corporation, Taiwan
Fang-Jing Wu	Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore
Zhenhui Jessie Li	Pennsylvania State University, USA

SDA Chairs

Jun Huan	University of Kansas, USA
Irwin King	The Chinese University of Hong Kong, Hong Kong
Michael R. Lyu	The Chinese University of Hong Kong, Hong Kong
Haiqin Yang	The Chinese University of Hong Kong, Hong Kong

DMDA-Health Chairs

Benyun Shi	Hong Kong Baptist University, Hong Kong
Kwok-Wai William Cheung	Hong Kong Baptist University, Hong Kong
Jiming Liu	Hong Kong Baptist University, Hong Kong

ALSIP Chairs

Koji Tsuda	AIST, Japan
Rajeev Raman	University of Leicester, UK
Shin-ichi Minato	Hokkaido University, Japan
Koji Tsuda	AIST, Japan

SocNet Chairs

Wookey Lee	Inha University, South Korea
Carson Leung	University of Manitoba, Canada
Cheng-Te Li	National Taiwan University, Taiwan
Ee-Peng Lim	Singapore Management University, Singapore
Guandong Xu	University of Technology, Sydney, Australia

DMBIH Chairs

Qian Zhu	Mayo Clinic, USA
Yuji Zhang	Mayo Clinic, USA
Hongfang Liu	Mayo Clinic, USA
Michael Lajiness	Eli Lilly and Company, USA

BigPMA Chairs

Keith C.C. Chan	Hong Kong Polytechnic University, Hong Kong
Jiun-Long Huang	National Chiao Tung University, Taiwan
Yi-Cheng Chen	Tamkang University, Taiwan

Combined Program Committee

Adam Dunn	University of New South Wales, Australia
Aiello Marco	University of Groningen, The Netherlands
Alejandro Lopez Ortiz	University of Waterloo, Canada
Alexander Lazovik	University of Groningen, The Netherlands
Alexander S. Gutfraind	University of Illinois at Chicago, USA
Aman Kansal	Microsoft Research, USA
Bagheri Ebrahim	Ryerson University, Canada
Bai Zhang	Johns Hopkins University, USA
Benjamin C.M. Fung	McGill University, Canada
Bin Guo	Northwestern Polytechnical University, China
Bo Zhao	Microsoft Research, USA
Bolin Ding	Microsoft Research, USA

Bouguettaya Athman	RMIT, Australia
Buqing Cao	Hunan University of Science and Technology, China
Chen Chen	Google Inc., USA
Cheng-Te Li	National Taiwan University, Taiwan
Chiara Renso	National Research Council (CNR), ISTI Institute in Pisa, Italy
Chih-Chieh Hung	Rakuten Inc., Japan
Chih-Hua Tai	National Taipei University, Taiwan
Christian Guttmann	IBM Research, Australia
Christopher Chute	Mayo Clinic, USA
Christos Efstratiou	University of Kent, UK
Chu-Cheng Hsieh	E-Bay Inc., USA
Cui Tao	University of Texas Health Science Center at Houston, USA
Dai Bing Tian	Singapore Management University, Singapore
David Buckeridge	McGill University, Canada
David Taniar	Monash University, Australia
David Wild	Indiana University, USA
Demetris Zeinalipour	University of Cyprus, Cyprus
Dimitrios Lymberopoulos	Microsoft Research, USA
Dingcheng Li	Mayo Clinic, USA
Dou Shen	Baidu Inc., China
Emilio Corchado	University of Burgos, Spain
Eric Gifford	Merck, USA
Erol Gelenbe	Imperial College London, UK
Fang-Jing Wu	Agency for Science, Technology and Research (A*STAR), Singapore
Fatos Xhafa	Universitat Politécnica de Catalunya, Spain
Feng Chen	Carnegie Mellon University, USA
Fengjun Li	University of Kansas, USA
Francisco Pereira	Singapore-MIT Alliance for Research and Technology, Singapore
Ganesh Kumar Venayagamoorthy	Missouri University of Science and Technology, USA
George Zhou	University of Wollongong, Australia
Giuseppe Ottaviano	University of Pisa and ISTI-CNR, Italy
Goce Trajcevski	Northwestern University, USA
Gordon Sun	Microsoft STC, China
Guanling Chen	University of Massachusetts Lowell, USA
Guanling Lee	National Dong Hwa University, Taiwan
Habtom W. Ressom	Georgetown University, USA
Hai Jin	HUST, China
Han-Wei Shen	Ohio State University, USA
Hiroki Arimura	Hokkaido University, Japan
Hongbo Deng	Yahoo! Labs, USA

Hoyoung Jeung	SAP Research, Brisbane, Australia
Ismail Khalil	Johannes Kepler University, Austria
James Bailey	University of Melbourne, Australia
Jennifer Jie Xu	Bentley University, USA
Jianhua Yao	Shanghai Institute of Organic Chemistry, China
Jiannong Cao	Hong Kong Polytechnic University, Hong Kong
Jiayu Zhou	Arizona State University, USA
Jing Gao	University at Buffalo, USA
Jinjun Chen	University of Technology, Syndey, Australia
Jintao Zhang	Adometry, USA
John S. Brownstein	Harvard University, USA
Joshua Zhexue Huang	Shenzhen University, China
Julia T.Y. Weng	Yuan Ze University, Taiwan
Jun Ma	Shandong University, China
Jyotishman Pathak	Mayo Clinic, USA
Kai (Kevin) Zheng	University of Queensland, Australia
Kamran Shafi	DSARC, UNSW, Australia
Kavishwar B. Wagholikar	Mayo Clinic, USA
Khalid Saeed	AGH University of Science and Technology, Poland
Korris F.L. Chung	Hong Kong Polytechnic University, Hong Kong
Kouroush Neshatian	University of Canterbury, New Zealand
Kun Zhang	Max Planck Institute for Intelligent Systems, Germany
Kunihiko Sadakane	National Institute of Informatics, Japan
Kun-Ta Chuang	National Cheng Kung University, Taiwan
Lean Yu	Chinese Academy of Sciences, China
Li Chen	National Cancer Institute, USA
Li Kuang	China Southern University, China
Lidong Bing	The Chinese University of Hong Kong, Hong Kong
Limin Zhu	ICT, Australia
Lin Hui	Tamkang University, Taiwan
Liqun Li	Microsoft Research Asia, China
Lu-An Tang	NEC Lab, USA
Lucila Ohno-Machado	University of California, San Diego, USA
Makoto Yamada	Yahoo! Labs, USA
Mao Ye	Klout Inc., USA
Mei Liu	New Jersey Institute of Technology, USA
Meng-Fen Chiang	Yahoo!, Taiwan
Meng-Shiuan Pan	Tamkang University, Taiwan
Michael Sheng	The University of Adelaide, Australia
Michela Antonelli	University of Pisa, Italy
Ming Li	Nanjing University, China

Mingdong Tang	Hunan University of Science and Technology, China
Mirco Musolesi	University of Birmingham, UK
Mi-Yen Yeh	Academia Sinica, Taiwan
Mohd Saberi Mohamad	Universiti Teknologi Malaysia, Malaysia
Mohyuddin	King Abdullah International Medical Research Center, Saudi Arabia
Nicholas Jing Yuan	Microsoft Research Asia, China
Parisa Rashidi	University of Florida, USA
Patricia Riddle	University of Auckland, New Zealand
Peter Oolog	Aalborg University, Denmark
Philip S. Yu	University of Illinois at Chicago, USA
Ping Luo	HP Labs, China
Qian (Jane) Yoo	Amazon, USA
Raghu K. Ganti	IBM T.J. Watson Research Center, USA
Rajarshi Guha	NIH Center for Advancing Translational Science, USA
Rajeev Raman	University of Leicester, UK
Redda Alhaj	University of Calgary, Canada
Richi Nayek	Queensland University of Technology, Australia
Ritu Chauhan	Amity Institute of Biotechnology, India
Robert Freimuth	Mayo Clinic, USA
Roberto Grossi	University of Pisa, Italy
Saeed u Rehman	Unitec, Institute of Technology, New Zealand
Seung-won Hwang	Pohang University of Science and Technology, South Korea
Shang Xia	National Institute of Parasitic Diseases, China CDC, China
Shangguang Wang	Beijing University of Posts and Telecommunications, China
Shin-ichi Minato	Hokkaido University, Japan
Shipeng Yu	Siemens Healthcare, USA
Shou-De Lin	National Taiwan University, Taiwan
Shusaku Tsumoto	Shimane University, Japan
Siddhartha R. Jonnalagadda	Northwestern University, USA
Stephen Chen	York University, China
Stephen Wu	Mayo Clinic, USA
Suh-Yin Lee	National Chiao Tung University, Taiwan
Susana Ladra	University of A Coruña, Spain
Tadashi Dohi	Hiroshima University, Japan
Takahiro Hara	Osaka University, Japan
Takeaki Uno	National Institute of Informatics, Japan
Tao Zhou	University of Electronic Science and Technology of China, China
Thomas Choi	Hong Kong Polytechnic University, Hong Kong
Ting Gong	MD Anderson Cancer Center, USA

Tsuyoshi Ide	IBM T.J. Watson Research Center, USA
Veli Makinen	University of Helsinki, Finland
Vinay Pai	NIH, USA
Wang-Chien Lee	Pennsylvania State University, USA
Wei Chen	Microsoft Research Asia, China
Wei Di	E-Bay Research, USA
Weike Pan	Shenzhen University, China
Wei-Shinn Ku	Auburn University, USA
Xiang Zhang	Case Western Reserve University, USA
Xiaohua Tony Hu	Drexel University, USA
Xiaoqian Jiang	University of California, San Diego, USA
Xiao-Zhi Gao	Aalto University, Finland
Xijin Ge	South Dakota State University, USA
Xin Li	Mayo Clinic, USA
Xin Wang	University of Calgary, Canada
Xing Xie	Microsoft Research Asia, China
Xue Li	University of Queensland, Australia
Xujuan Zhou	University of New South Wales, Australia
Xumin Liu	Rochester Institute of Technology, USA
Yan Huang	University of North Texas, USA
Yan Shen	Queensland University of Technology, Australia
Yanchang Zhao	RDataMining.com, Australia
Yanchun Zhang	Victoria University, Australia
Yang Xiang	Deakin University, Australia
Yanjun Yan	Western Carolina University, USA
Yanping Xiang	University of Electronic Science and Technology of China, China
Yao Zhao	Beijing University of Posts and Telecommunications, China
Yi Cai	South China University of Technology, China
Yitan Zhu	NorthShore University HealthSystem, USA
Yizhou Sun	Northeastern University, USA
Yizhou Sun	University of Illinois at Urbana-Champaign, USA
Yongkun Wang	Rakuten Inc., Japan
Youngki Lee	Singapore Management University, Singapore
Yu Zheng	Microsoft Research Asia, China
Yutao Ma	Wuhan University, China
Zhenglu Yang	University of Tokyo, Japan
Zhiang Wu	Nanjing University of Finance and Economics, China
Zhijun Yin	Twitter, USA
Zhixian Yan	Samsung Research, USA

Contents

Data Analytics for Targeted Healthcare

Data Mining and Decision Analytics for Public Health and Wellness

Biologically Inspired Techniques for Data Mining

Mobile Data Management, Mining, and Computing on Social Networks

Big Data Science and Engineering on E-Commerce

Cloud Service Discovery

Mobile Sensing, Mining and Visualization for Human Behavior Inference

Scalable Data Analytics: Theory and Applications

Algorithms for Large-Scale Information Processing in Knowledge Discovery

Data Mining in Social Networks

Data Mining in Biomedical informatics and Healthcare

Pattern Mining and Application of Big Data

Data Analytics for Targeted Healthcare

Conversation Intention Perception Based on Knowledge Base

Yi-Zheng Chen(✉), Hua-Kang Li, and Yi Liu

Shanghai Key Laboratory of Scalable Computing and Systems,
Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, China
c.yizheng@sjtu.edu.cn,
{huakang.lee,ly0406}@cs.sjtu.edu.cn

Abstract. Web Intelligence is gaining its growth in a rapid speed. The notion of wisdom, which is considered as the next paradigm shift of WI, has become a hot research topic in recent years. The basic application of wisdom is making a short conversation in an interactive and understandable way based on the huge web resources. However, current conversation system normally applies the recognition of semantic similarities in the prepared database, neglecting the true intention hiding in the expression. In this paper, we present a model based on the medical Q&A knowledge base to overcome this challenge. The knowledge base includes three parts: disease entity, medicine, properties. A simple graph path algorithm based on words direction and relation weight adjustment is used to realize conversation intention perception. The experimental results show that this method can effectively perceive types of intention. This method can also be applied in deep understanding of other intelligent systems such as classifications and text mining.

Keywords: Web Intelligence · Intention perception · Knowledge base · Conversation system · Graph path

1 Introduction

Web Intelligence (WI) is a new direction of academic research and industry development. The main duty of WI is making use of various web information and knowledge in a professional and effective way based on technologies such as knowledge discovery, data mining, intelligent agents as well as advanced information technology [1]. In the area of WI technology, the notion of wisdom [2] is gaining much attention in scientific research. In a simple practice, the concept of wisdom contributes to the conversation system, in which person and computer act in an unobstructed and easy manner just like the communication between human beings. This application needs to grasp the real intention of human's sentences accurately and comprehensively, which means to understand and know what his true demand is.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 3–14, 2014.
DOI: 10.1007/978-3-319-13186-3_1

One traditional conversation system is to measure the semantic similarities between human inputs [3], which is not trivial to realize but the performance is unsatisfied when the input has a little word overlap. In addition, the implication of conversation always depends on the keyword's assembly among sentences. For example, about the health care consult, a patient mentioned the symptom or his information in a pretty long sentence. After all, he made clear name of medicine and want to know whether the medicine is beneficial for curing the disease, or whether it may bring side effect to his current condition. The traditional conversation system would determine the patient is talking about disease according to the symptoms and would recommend the most common treatment. The problem is that these systems cant identify customer's real intention from the given information.

The most popular online medical answering or guiding systems are mainly relied on manual consult in China. Health care system and the modern health infrastructure play an essential role in recent years [4]. However, self-management for health care has two challenges: (a) building a health knowledge base with comprehensive diseases, medicine information automatically. In recent years, more and more knowledge bases with massive data are building up, such as Wikipedia[1], Wordnet[2], Baike[3] and so on. Most of these knowledge bases are established by manually editing. (b) developing an intelligent consulting system which could detect customer's intention and provide some treatment recommendations within a short conversation. The applications of knowledge base for WI are still very few.

In this paper, we use massive health care Q&A data to build a health knowledge base and develop a conversation intention perception system in Chinese. The knowledge base includes three parts: disease entities, medicine entities and symptom properties. The associated relation links between them are created according to a simple graph path algorithm. We use a content center detection algorithm based on the knowledge graph to estimate the conversation intention. The experimental results show that this method can effectively perceive requirement types. This method can also be applied in deep understanding of other intelligent systems such as classifications and text mining. The main contributions of this paper are outlined as follows:

- Based on medical entities, we extracted disease entities, medicine entities, symptom entities from online resources using keyword extraction and feature selection method.
- According to the associated relations between keywords in a sentence, we proposed an automatic knowledge base building approach. We extend the association relation between entities nodes in the built knowledge base to construct the relation map and weights between nodes.
- According to the knowledge graph path and relation weight, we identify the conversation intention within a short conversation.

[1] http://www.wikipedia.org/

[2] http://wordnet.princeton.edu/

[3] http://www.baike.com/

The main organization of this paper is listed as follows. Section 2 discusses the most related works, including stat-of-the art approaches on intention perception and traditional conversation system. We describe the data collection and knowledge base construction in Sect. 3. The intention perception algorithm is explained in Sect. 4. Section 5 illustrates the experimental performance and evaluates several factors, which may affect the performance. We summary the paper with discussion on future work in Sect. 6.

2 Related Work

Conversation system aims at finding similar context in existing data set. Earlier works mainly focus on using the semantic similarities between sentences such as the overlap coefficient, Dice coefficient and Jaccard coefficient to get the desirable result [5]. To solve the problem that the above methods work poorly when there is little word overlap between queries, latter researches have achieved big progress using the statistical techniques of information retrieval. Jeon et al. [6] study automatic methods of finding semantically similar question pairs based on the assumption that similar answers lead to approximate questions. Ko et al. [7] apply answer relevance and answer similarity into the statistical model, and he made an improvement to this model considering correlation of the correctness of answer candidate [8]. These systems mainly rely on the semantic similarities of human inputs and neglect the user's intention implicated in them.

Intention perception is the key technology for the conversation system since the understandable machine performs well returning the answer [9]. It is a tough work considering the various human actions, and most of its researches are applied in the academic field of human-robot [10]. One of the main obstacles is that user's intention recognition contains the uncertainties, and Jeon et al. [11] proposes an ontology-based approach to minimize them. Some other research works apply the machine learning method to solve the issue. Kuan et al. [12] use the Support Vector Machine(SVM) and Linear Regression as two steps to identify human intention. Hofmann et al. [13] adopt the Bayesian belief networks to form the intention model. These methods have been proved effective in their domain.

Although context semantic and machine learning approaches have good performance in simple dialogue, it is still very difficult to deal with the new knowledge growing. On the other hand, the context-based conversation intention approach can't associate the current knowledge with linked or similar knowledge as humans. Therefore, we use a knowledge base as the fundamental element to attempt conversation intention perception.

3 Knowledge Base Building

Now, several common and large knowledge bases such as Wikipedia, MozillaZine[4], Probase [14], GeoNames[5] and WordNet [15] have been set up manually

[4] http://kb.mozillazine.org/Knowledge_Base

[5] http://www.geonames.org/

or semiautomatically. Here we use massive Q&A data set[6] to build a content based knowledge base. We use distributed web clawer to download target web page and tools like DOMTree to translate the gained Q&A information into the XML format.

3.1 Q&A Archive

Table 1. Structure of question and answer pair

URL	http://120ask.com/question/34672281.html
Question Title	Body itch
Question Body	When the summer comes, my body itch and exists red dot...
Requirement	What disease it is
Answers	It may relate to allergy which is caused by summer insects...

The Q&A archive we collected are organized in Table 1. These Q&A pairs in our experiment are all Chinese. In this paper, we translate the Chinese words into English to make our examples more clear. Each item in archive has 5 fields: URL part is the unique identifier of question and answer pair. Question Title is the short description for the question and Question Body gives a detail statement about question. It is the basic data for our experiment. The average length of question body is 48 words in Chinese. Part Requirement represents the kind of help questioner is looking for. Therefore, this part contains the standard for questions' intention classification. The last part answers is the selected answer from several candidates for the corresponding question and its average length is 108 words in Chinese.

We collected 30 million Q&A pairs from the web and divided them into two collections: 25 million pairs for the training data and 5 million for the testing data. We use the requirement part to mark the training data and testing data. Phrases such as "what disease" or "how to cure" or "what medicine" or "negative influence" in the requirement part are applied to mark the question to its corresponding category. As not all the people fill in the requirement part, finally we receive nearly 1 million marked training data and 200 thousand marked testing data for our experiment. In the preprocess procedure of data set, we remove some redundant words from the questions, such as stop-words, digits and links.

3.2 Knowledge Base

To build up the knowledge base for our experiment, we need to collect three types of medical entities: disease entity, medicine entity and symptom entity. And then the relation between them is formed. The established knowledge base is the preparation and fundamental element for the next stage of experiment.

[6] http://www.120ask.com

- Disease and Drug Entities: The former two entities are professional words and obtained by web crawling. Baidu Encyclopedia (BE)[7] is an open content online encyclopedia which covers all areas of knowledge in Chinese. The Maximum entropy classifier is adopted to classify those entries into large mount of categories using structural information since the BE pages are well tagged. After receiving the labeled entities, we get nearly 25000 disease entities and 9800 medicine names.
- Symptom Entity: Since most symptom entities are not professional words and happen in the oral presentation, they can not be easily and accurately discovered from the professional encyclopedia web sites. We extract symptom entities from the collected question and answer pairs based on the assumption that most symptoms appear many times in oral presentation since patients usually have limited words to describe their diseases. Thus we extract the phrases exist frequently in question and answer pairs and then combine the phrases with the adverbs of positive and negative words. After the artificial selection we get nearly 3000 symptom entities.
- Relation Map: For the knowledge base building several relationships are identified: diseases have corresponding symptoms, diseases can be cured by corresponding medicine, symptoms can be cured by corresponding medicine. Thus, the Q&A pairs are used since the relationship of entities is hiding in them. We assume that the more frequently entities appear simultaneously in the Q&A pair, the more likely they are connected. The bigger frequency is, the closer their relationship is. After the filter process, we build up the relation map among these entities.

4 Intention Perception

The basic assumption of our model is using the medical knowledge base and relation map to adjust the keywords weights of different category intention based on correlative strength and graph path. We assume that the entity which receives more connections from other entities is more important in the conversation. Therefore, the more entities connected to the current phrase, the more weight value will be added to the current phrase.

The intention perception problem is actually a dynamic classification problem. We divide the medical questions into four types of intention, they are listed as follows:

- askers are willing to know what disease it may be
- askers are willing to know how to cure the disease or the described symptom
- askers are willing to know the medicine to cure the disease or the symptom
- askers are willing to know the negative influence of mentioned medicine

[7] http://baike.baidu.com/

4.1 Weight Adjustment

The relation map of entities based on the knowledge base we established before is shown in Fig. 1. The double-ended arrow represents the two entities are connected directly, and the digit stands for the co-occurrence frequency. Firstly, we compute the distance between two entities. For example, distance between entity Gastritis and entity Vomit is considered as one as they are connected directly, distance between entity Gastritis and entity Fracture is two since they are connected through entity Pain, while there are three units of distance between entity Gastritis and entity Bleeding. And the distance of two entities has a limitation of four. Two entities are connected within the shortest path. Secondly, when a query is given, we use Jieba Participle[8] to depart the question into phrases. The initial weight of each phrase is endowed as one. Thirdly, to a certain entity, the direct and indirect connected entities make a contribution to its weight value, we call it the contribution value. The closer distance and bigger co-occurrence of two entities both devote to larger contribution value. The formulation to compute contribution value is as follows:

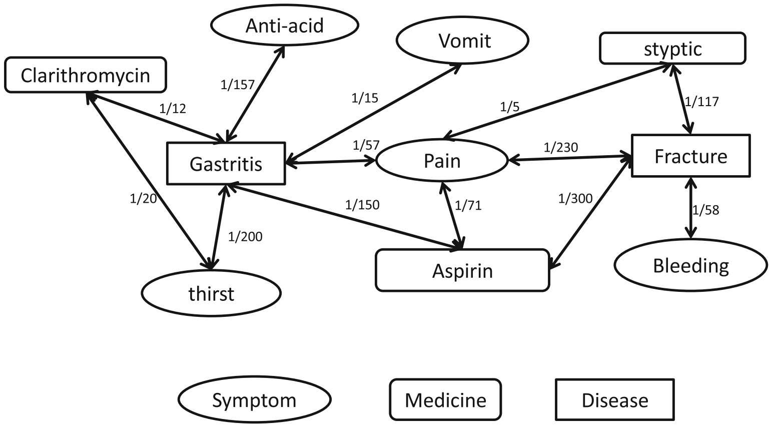


Fig. 1. The relation map of entities.

$$\left(1-\frac{1}{\log F_e}\right) * Y_X \qquad (X \times Y \in R,\ R = <1,a>,<2,b>,<3,c>,<4,d>) \qquad (1)$$

where F_e is the co-occurrence frequency of two entities, Y_X is the initial contribution value to the corresponding distance value X.

Considering the fact each kind of entities stands for different character of given conversation, for example, in the situation of medical system, although a symptom entity and a medicine entity both connect to a disease entity directly,

[8] https://pypi.python.org/pypi/jieba/

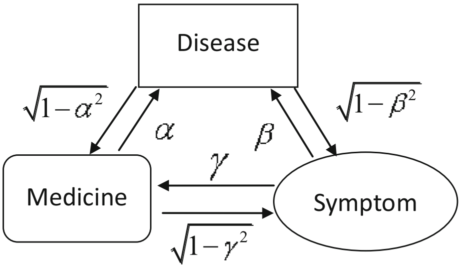


Fig. 2. The contribution multiplier between connected entities.

their influence to the weight of disease entity is not the same, we call this influence the contribution multiplier. Figure 2 shows the contribution multiplier we settled in our model. For instance, the contribution multiplier of medicine to disease is α, then in turn the contribution multiplier disease to medicine is $\sqrt{1-\alpha^2}$.

When come to the situation that two entities are from the same category, their distance is at least two as they can not be connected to each other directly. Their contribution multiplier is as follows:

$$contributionmultiple(A_1, A_m) = \prod_{i=1}^{m-1} contributionmultiple(A_i, A_{i+1}) \tag{2}$$

where A_1, A_m stand for the two entities from the same category, they are connected by the path from A_2 to A_{m-1}.

Combining the contribution value and contribution multiple, we set up computational method of entity weight, it is shown as follows:

$$weight(w_i) = initialweight + \sum_{j=1,j!=i}^{n} contibution\ value * contribution\ multiple \tag{3}$$

4.2 Language Model

Language Model (LM) can be either probabilistic or non-probabilistic. The probabilistic language model is widely used in the field of data mining and natural language processing. In this paper, we adopt a probabilistic model to complete the classification task. First, we estimate the probability the subsequence of words relate to the category. Then rank the probability value and deem the category which has the highest probability is the one this question belong to.

We use $c1, c2, c3, c4$ to denote different types of intention. In order to classify the given question to which category, we need to get the question likelihood computed by $P(q|c)$, and the formulation is as follows:

$$P_r\left(q|c\right) = \sum_{i=1}^{N} P_r(w_i|c) \tag{4}$$

where N represents the number of words in the query and $P_r(w_i|c)$ stands for the probability word w_i occurs in current category c. The formulation we use is a multinomial distribution which indicates that the distribution of each phrase in the question is generated independently, they obey the same probability distribution. In order to compute $P_r(w_i|c)$, we assemble all the questions from the same type to one synthetic document. Then the maximum likelihood estimate (MLE) is adopted which computes the probability as follows:

$$P_r\left(w_i|c\right) = \frac{F_{ic}}{F_c} \tag{5}$$

In the formulation, the F_{ic} represents the count of training data items which w_i exits and F_c is the size of training data of the current category.

The probability of words need to be normalized to make the sum of them in all the categories to be one. Let $S_u = \sum_{i=1}^{V} P(w|c_i)$ be the normalization factor, then we recalculate the probability of words in the following rule:

$$P_r(w|c_i) = \frac{P_r(w|c_i)}{S_u} = \frac{P_r(w|c_i)}{\sum_{i=1}^{V} P_r(w|c_i)} \tag{6}$$

4.3 Combined Model

Our model combines the weight we have endowed to each phrase in the given conversation and the probability language model. The weight represents the importance of each attribute to the classification. In other words, a word with higher weight contributes more than others to the probability estimation in the classification [16]. The formulation in our model is shown as follows:

$$P_r\left(q|c\right) = \frac{\sum_{i=1}^{N}(weight(w_i) * P_r(w_i|c))}{\sum_{i=1}^{N} weight(w_i)} \tag{7}$$

5 Experimental Results

5.1 Data Set

In the data preparing process, we collect nearly 1 million training data and 200 thousand testing data to evaluate the proposed method. Actually, the data set for the four types of conversation is not evenly distributed, especially for the fourth category which people are looking for the negative of medicine. The detail of training data corpus is shown in Table 2. As for the testing data corpus, to be even, we equally divided it into three testing data sets, each contains 2000 articles for each category and 8000 pairs for the whole. In the later experiment, we will use these three data sets to make a comparison to ensure our model's performance.

Table 2. Training data corpus

Category	Type1	Type2	Type3	Type4	Total
Data size	165422	454036	240580	53036	913074

5.2 Evaluation Measure

In our experiment, we compare our method with two basic methods, BOOL and TF-IDF. These three methods included all rely on the Eq. (7), while they differ from each other the weight value in the equation. The BOOL method treats each phrase in the sentence equally. Thus the weights of phrases are all be endowed as one. Method TF-IDF is very common. It uses the term frequency and inverse category frequency value of words as its weight value. Speaking of the evaluation metrics, accuracy is adopted which is commonly seen in the field of data mining and statistics. Accuracy is a measure of the percentage that the testing data is correctly classified.

5.3 Comparison of Methods

To adopt our method, since some parameters are involved in the equation we first need to give some certain value to these parameters. In this paper, the initial contribution value $[a, b, c, d]$ is fixed and regarded as $[0.75, 0.5, 0.25, 0]$. While parameters α, β, γ are variable in the range of $[0, 1]$. Later we will adjust these variable parameters to make our model better suit to intention perception in the question.

In the given testing data, as the incomplete of knowledge base we have established, it is a fact that there might be no entity in the knowledge base can be found in the question or the found entities have no connection between each other. Facing these situations, our model will give each word the weight of one just as the BOOL method does. To see how our method works in the testing data which only involves connected entities, we remove the testing items which contain the above features and finally get nearly 11 thousand testing data pairs to form the fourth testing corpus. Thus the four testing corpus we use are as follows: the former three each contains 8 thousand items and the fourth one contains items which have connected entities.

Figure 3 shows the comparison of three methods applied in the four different testing data corpus. In the experiment, the parameters α, β, γ are 0.7,0.9,0.9 respectively since they achieved the best result after several tests. From the figure we find that the TF-IDF method works no better than BOOL method which is reasonable as TF-IDF method is not effective in the keyword extraction when the sentence is short. While our model performs much better than these two methods especially when the data set only contains the questions which have connected entities as shown in the fourth histogram. It proves that the method we proposed can effectively grasp the central topic of question and get to know people's intention more accurately than the other two methods.

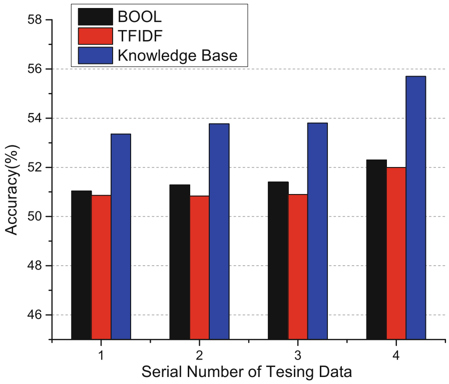


Fig. 3. The comparison of three methods in the human's intention understanding

5.4 Parameters Evaluation

As mentioned before, the model we adopted in this paper has a fixed set as a, b, c, d while α, β, γ are variable to optimize intention perception results. Figures 4, 5 and 6 demonstrate how the three parameters influence the accuracy of intention perception work. In every figure, the other two parameters are fixed to a static value as 0.7, so that the contribution multiple between each other is the same. It implies that the entities of medicine or disease tend to receive bigger contribution multiple parameter compare to entities of symptom which is rational since they exist less frequently than entities of symptom in a single question. Thus the former two kinds of entities are more presentative and should get a bigger contribution multiplier.

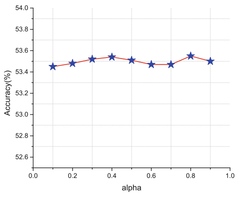


Fig. 4. The influence of alpha Parameter

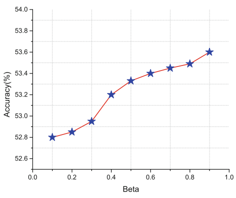


Fig. 5. The influence of beta Parameter

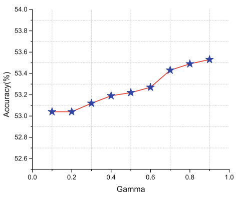


Fig. 6. The influence of gamma Parameter

5.5 Sentence Length Effect

As we know, most sentences in conversation system are short. The number of keyword still fluctuates within a certain range. It is meaningful to measure how the three methods work when the number of words in question ranges in a given interval. We divide the testing data according to their length by steps of 20 words. From Fig. 7, we easily discovery that our method performs better than the other two when the number of words are neither too small nor too big. The small one devotes to limited number of entities while the big one contains too much information which easily makes some words over-weighted. The performance of three methods were all very low because it's difficult to extract entities. While the sentence length is over 100, the increase is not so significant for knowledge base.

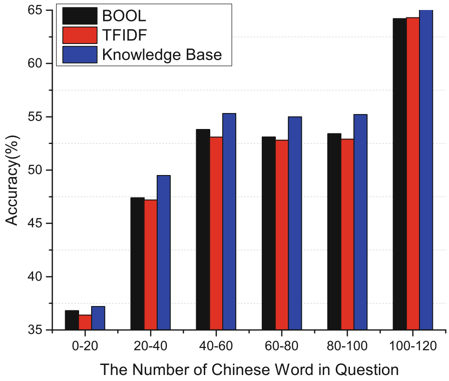


Fig. 7. The comparison of three methods in question of different length

6 Conclusion

In this paper, we crawled massive health conversation content to build a health care knowledge base. After word segmentation, keywords were extracted and symptom entities were selected using the feature candidate algorithm. The health care knowledge was built based on the association relation between diseases, medicine and symptom entities. We proposed a simple graph path and weight calculation algorithm to modify the association relation and transmission weights to estimate the intention center words. We used a Bayesian model to estimate the customers intention within short content conversation. Finally, we illustrated several experimental results with effectively perceived intention types.

Since the real conversation system likes a catch ball game, we will devote this model to build an interactive dialogue system. Furthermore, we would introduce living place, hospital name and age stage to enrich the knowledge base. And this method would be extended to other content areas such as travel consult and social network.

References

1. Liu, J., Zhong, N., Yao, Y., Ras, Z.W.: The wisdom web: new challenges for web intelligence (wi). J. Intell. Inf. Syst. **20**(1), 5–9 (2003)

2. Li, J., Zhang, K.: Keyword extraction based on tf/idf for chinese news document. Wuhan Univ. J. Nat. Sci. **12**(5), 917–921 (2007)
3. Jeon, J., Croft, W.B., Lee, J.H.: Finding similar questions in large question and answer archives. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, pp. 84–90. ACM (2005)
4. Kunz, H., Schaaf, T.: General and specific formalization approach for a balanced scorecard: An expert system with application in health care. Expert Syst. Appl. **38**(3), 1947–1955 (2011)
5. Manning, C.D., Schütze, H.: Foundations of statistical natural language processing, vol. 999. MIT Press, Cambridge (1999)
6. Jeon, J., Croft, W.B., Lee, J.H.: Finding semantically similar questions based on their answers. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 617–618. ACM (2005)
7. Ko, J., Si, L., Nyberg, E.: A probabilistic framework for answer selection in question answering. In: HLT-NAACL, pp. 524–531 (2007)
8. Ko, J., Nyberg, E., Si, L.: A probabilistic graphical model for joint answer ranking in question answering. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 343–350. ACM (2007)
9. Zhang, Y., Wang, X., Wang, X., Fan, S., Zhang, D.: Using question classification to model user intentions of different levels. In: IEEE International Conference on Systems, Man and Cybernetics, SMC 2009, pp. 1153–1158. IEEE (2009)
10. Awais, M., Henrich, D.: Online intention learning for human-robot interaction by scene observation. In: 2012 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), pp. 13–18. IEEE (2012)
11. Jeon, H., Kim, T., Choi, J.: Ontology-based user intention recognition for proactive planning of intelligent robot behavior. In: International Conference on Multimedia and Ubiquitous Engineering, MUE 2008, pp. 244–248. IEEE (2008)
12. Kuan, J.-Y., Huang, T.-H., Huang, H.-P.: Human intention estimation method for a new compliant rehabilitation and assistive robot. In: Proceedings of SICE Annual Conference 2010, pp. 2348–2353. IEEE (2010)
13. Hofmann, M., Lang, M.: Intention-based probabilistic phrase spotting for speech understanding. In: Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2001, pp. 99–102. IEEE (2001)
14. Wu, W., Li, H., Wang, H., Zhu, K.Q.: Probase: a probabilistic taxonomy for text understanding. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 481–492. ACM (2012)
15. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: Yago2: a spatially and temporally enhanced knowledge base from wikipedia. Artif. Intell. **194**, 28–61 (2013)
16. Zhang, H., Sheng, S.: Learning weighted naive bayes with accurate ranking. In: Fourth IEEE International Conference on Data Mining, ICDM'04, pp. 567–570. IEEE (2004)

An Empirical Algorithm for Bias Correction Based on GC Estimation for Single Cell Sequencing

Bo Xu, Tengpeng Li, Yi Luo, Ruotao Xu, and Hongmin Cai(✉)

School of Computer Science and Engineering,
South China University of Technology, Guangzhou, China
hmcai@scut.edu.cn

Abstract. Whole genome amplification (WGA) have been applied to single cell copy number variations (CNVs) analysis, which is a common genomic mutation associated with various diseases and provides new insight for the fields of biology and medicine. However, the WGA-induced bias based on multiple displacement amplification (MDA) significantly limits sensitivity and specificity for CNVs detection. To address the limitations, an empirical algorithm for CNVs detection at single cell level was developed. This proposed method consists of base call amplification, alig- nment and analysis to remove the MDA-induced bias. **We generated and analyzed about 50G short read data sets based on MDAsim, a software to amplify the chromosome 21 into various coverage**. Simulation experiments have shown **that the coverage tended to be less than average in genomic GC-enriched (>45 %) regions, implying a significant amplification bias within these regions. Base substitution error frequencies with G > A transversion is being among the most frequent and C > T, G > T transversions are among the least frequent substitution errors. The estimated substitution was employed to compensate errors to correct bias readings.**

Keywords: Amplification bias · Substitution error · GC correction

1 Introduction

Whole genome amplification (WGA), a widespread approach to amplify inadequate amounts of DNA samples for sequencing in Single-cell genomics analysis [1–3] have been extensively used to single-cell copy number variations (CNVs) analysis, at the cost of introducing biases [4–8]. As a key factor in cancer mutation [1, 9], CNVs is a common genomic variation closely associated with assorted diseases, the detection and analysis of which contributes to the research of biology and medicine.

Limited by the number of the specimens, WGA methods are widely used to facilitate the CNVs detection and analysis at single-cell level, such as Polymerase chain reaction (PCR) and multiple displacement amplification (MDA). Multiple displacement amplification (MDA), a DNA amplification method widely used in Single-cell genomics studies, uses Φ29 DNA polymerase and random primers to generate large amount of DNA template for genome samples [10]. Compared with PCR-based

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 15–21, 2014.
DOI: 10.1007/978-3-319-13186-3_2

amplification method, MDA can be amplifid to the output with high quality and low error rates while not limited by the target length [10].

The introduction of WGA method insures the accuracy of CNVs' detection, nevertheless, it at the same time gives rise to amplification biases [4–8]. Although the mechanism of how the DNA polymerase function is influenced by GC content remains unsettled, it has been suggested that the amplification quality of template is closely associated with GC content [5]. The over-amplification or under-amplification of specific region of template can result from the rich or poor GC content [5], causing misrepresentation of that region. Thus, the WGA-induced bias significantly limits sensitivity and specificity for CNVs detection.

To investigate the limitation, an empirical algorithm was developed for CNVs detection at single cell level. The proposed method consists of base call amplification, alignment and analysis for MDA-induced bias removal, with the aid of Multiple Displacement Amplification Simulator (MDAsim). MDAsim is a software developed to simulate MDA process, which generates simulated reads well approximated to the experimental ones [11]. By comparing the simulated outputs with the input chromosome 21, corrective measures were carried out to remove and compensate the MDA-induced biases. The proposed algorithm is expected to optimize the MDAsim analysis and improve the accuracy of the simulation process.

In the proposed algorithm, chromosome 21 from human genome was selected as reference template and was amplified into various coverage based on MDAsim, thus generating about 50G short read data sets. Each read has been trimmed to 50 bases and aligned to chromosome 21 by BWA [12]. Extensive statistical analysis has been conducted to investigate the correlation between genomic GC content and corresponding read coverage, per-positon error numbers considering the wrong base calls only, per-base error rate considering all base calls.Finally, we conclude the base substitution error frequencies.

2 Methods

A systematical pipeline was designed to analyze the simulated data set. The pipeline consists of three steps: amplification, alignment and analysis. The chromosome 21 was selected to amplify its base calls by MDAsim [11].

Step 1: Amplification. Since the whole chromosome 21 is too large to analysis by the amplification software. The 48 M reference was splitted into 45 subgroups, each of those is 3 M in length with the index repeating 2 M each time (1–3, 2–4……). The resulted 3 M fasta file was then used to amplify the chromosome 21. With the help of MDAsim [11], chromosome 21 is amplified into different coverage range under various parameter settings to simulate the reads with different GC contents.

Step 2: Alignment. BWA [12] is used to map the amplified reads in different coverage against the reference template. Its alignment process generates the intermediate binary sai file and final sam file. In the sam file, BWA outputs the sam file in the SAM format [14], each line of which consists of the alignment information of each read.

Step 3: Analysis. To extract the classified errors from the BWA outputs and analyze the MDA-induces biases, an extensive statistical analysis has been developed to analyze the correlation between the read coverage and GC content, base substitution errors in reads, per-postion error numbers considering the wrong base calls and per-base error rate considering all the base calls.

3 Results

The chromosome 21 was amplified into different coverage, extending from 40 to 60. The BWA analysis was then conducted on the resulted data sets.

Because only in that coverage can we find the output with U1 (match with exactly one error(insertion or replacement).Finally, we acquired 90923032 50mer reads from the process that the perl scripts reported to be uniquely matched against the chr21 reference sequence which were labeled U0, U1 or U2 respectively (Fig. 1).

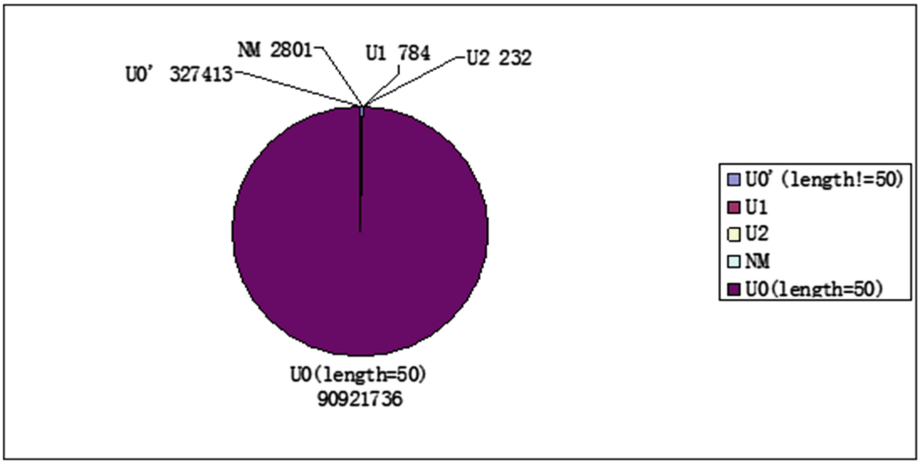


Fig. 1. Pie chart of the read analysis. The four categories are NM, no match found; U0, exact match found without any error; U1, match with exactly one error (insertion or replacement); U2, match with exactly two errors (insertion or replacement); U0', exact match found without any error, but its length is less than 50.

- **Correlation between the read coverage and GC content**

  The amplification was amplified and aligned with the coverage 50 from the chromosome 21 to analyze the GC biases in WGA. The number of reads starting in a sliding window of length in 1kbp is estimated firstly. The analysis of the correlation between the statistic and the characteristic of the sequence of chromosome 21 shows a positive correlation between the read coverage and GC content. The coverage increases as well as GC content. However, when GC content is larger than 45 %, the coverage decreases with the GC content.increasing.

  We defined the quotient between the reads number of each observation window and the average reads number as relative read number (RRN) [13], which ideally

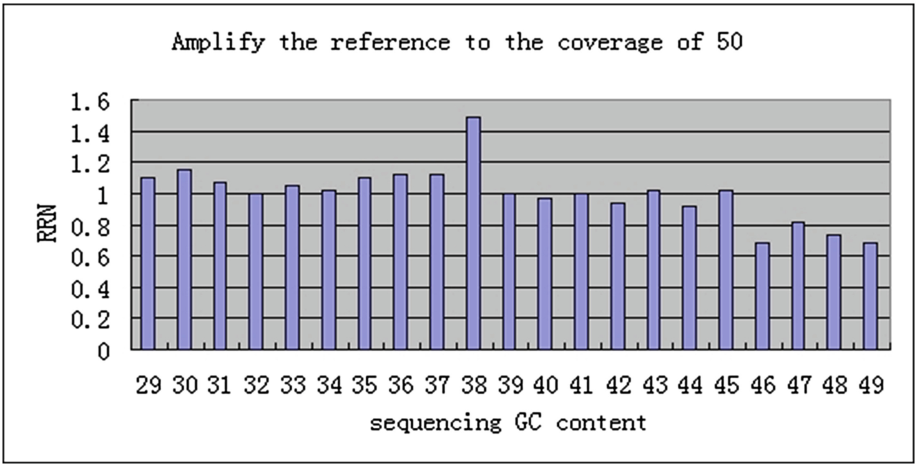


Fig. 2. Correlation of the read coverage and GC content: 50mer reads acquired from the chromosome 21. Each bar corresponds to the number of reads recorded for a 1-kbp window.

Table 1. Base substitution frequencies in the read data sets

Into\From	G	A	C	T	Any
G	–	0.01	0.05	0.04	0.1
A	0.49	–	0.15	0.12	0.76
C	0.04	0.02	–	0.04	0.1
T	0.01	0.02	0.01	–	0.04
Any	0.54	0.05	0.21	0.20	–

would be equal to one. By comparing the GC content and RRN, we discovered that the RRN tended to be less than average in genomic GC-rich (>45 %) (shown in Fig. 2), implying the amplification bias within these regions. Futhermore, the base substitutional analysis was done in these regions to correct the biases.

- **Analysis of base substitution errors in reads**
  The overall substitution error is calculated and summarized in Table 1. There are twelve possible substitution errors (8 transversions and 4 transitions) when a base call happens. The transition error of G > A happens most frequently, which accounts for almost half of the substitution errors, and the least frequent substitution error is G > T and C > T. The most frequent base to happen substitution error is G, and the least is A. However, A is the most frequent base to be changed into while T is the least.

  Futher experiment is also done to analyze the GC-enriched (>45 %, Fig. 2) region's substantial error, through which we can compensate the biased region' (>45 %, Fig. 2) substitutional base call. The transition error of T > C happens most frequently, which accounts for almost half of the substitution errors, and the least frequent substitution error is T > A and G > C. The most frequent base to happen

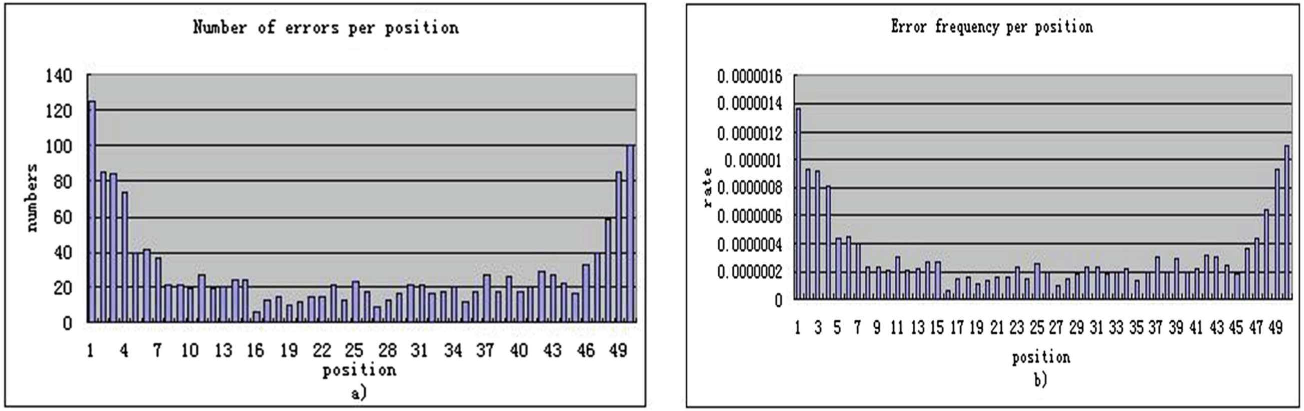


Fig. 3. Numbers of wrong base calls in reads depending on the position along the read. (a) Per-positon error numbers considering all the wrong base calls. (b) Per-base error rate among all the base calls.

substitution error is T, and the least is A. However, A is the most frequent base to be changed into while G is the least. With these estimated substitutional information, we compensate errors to correct bias readings in the GC-enriched (>45 %, Fig. 2) regions.

- **Numbers of wrong base calls in reads verses the position along the read**
  All the U1 U2 and U3 reads are selected for analysis, i.e. 3817 read (cf. Fig. 1), on the occurrence of errors per position. Two types of measurements are provided to quantify the errors. The first measurement calculated per-positon error numbers considering all the wrong base calls. The second measurement calculated per-base error rate among all the base calls. The results are shown in Fig. 3. The figure (a) shows that the high fraction of the wrong base calls occurs at the first and last position of the read. 8.2 % of the errors in the data sets are found at read position 1, and 6.7 % of errors are found at the last read position (position 50 in the data set Fig. 3a). The rate of the wrong base calls (Fig. 3b) has shown similar tendency. The rate is the highest at the first position along the read and the second highest at the last position of the read.

4 Conclusion

In this study, an algorithm was developed to detect the bias of multiple displacement amplification and the relation between GC content and coverage at the single cell level. The proposed method consists of base call amplification, alignment, analysis and base call substitutional compensate. The chromosome 21 was selected and amplified into 50 coverage. The defined RRN shows that the coverage tends to be less than average within GC-rich regions (Fig. 2). The GC-rich regions' substitution error and overall substitution error were extensively analyzed and estimated to compensate the base substitution error. For the overall reads, wrong base calls are frequently preceded by base G. Base substitution error frequencies vary with G > A transversion being among the most frequent and C > T, G > T transversions among the least frequent substitution errors. With these estimated substitutional information, we compensate errors to correct bias readings in the GC-enriched (>45 %, Fig. 2) regions. For the biased region (GC-rich regions), the transition error of T > C happens most frequently, and the least frequent substitution error is T > A and G > C. With these estimated substitutional information, we compensate the errors to correct the MDA-induced bias.

References

1. Navin, N., Kendall, J., Troge, J., Andrews, P., Rodgers, L., McIndoo, J., Cook, K., Stepansky, A., Levy, D., Esposito, D.: Tumour evolution inferred by single-cell sequencing. Nature **472**(7341), 90–94 (2011)
2. Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R.: The sequence alignment/map format and SAMtools. Bioinformatics **25**(16), 2078–2079 (2009)

3. Benjamini, Y., Speed, T.P.: Summarizing and correcting the GC content bias in high-throughput sequencing. Nucleic Acids Res. **40**(10), e72 (2012)
4. Dohm, J.C., Lottaz, C., Borodina, T., Himmelbauer, H.: Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. Nucleic Acids Res. **36**(16), e105 (2008)
5. Voet, T., Kumar, P., Van Loo, P., Cooke, S.L., Marshall, J., Lin, M., Esteki, M.Z., Van der Aa, N., Mateiu, L., McBride, D.J.: Single-cell paired-end genome sequencing reveals structural variation per cell cycle. Nucleic Acids Res (2013)
6. Hou, Y., Song, L., Zhu, P., Zhang, B., Tao, Y., Xu, X., Li, F., Wu, K., Liang, J., Shao, D.: Single-cell exome sequencing and monoclonal evolution of a JAK2-negative myeloproliferative neoplasm. Cell **148**(5), 873–885 (2012)
7. Talkowski, M.E., Rosenfeld, J.A., Blumenthal, I., Pillalamarri, V., Chiang, C., Heilbut, A., Ernst, C., Hanscom, C., Rossin, E., Lindgren, A.M.: Sequencing chromosomal abnormalities reveals neurodevelopmental loci that confer risk across diagnostic boundaries. Cell **149**(3), 525–537 (2012)
8. Dean, F.B., Nelson, J.R., Giesler, T.L., Lasken, R.S.: Rapid amplification of plasmid and phage DNA using phi29 DNA polymerase and multiply-primed rolling circle amplification. Genome Res. **11**(6), 1095–1099 (2001)
9. Tagliavi, Z., Draghici, S.: MDAsim: A multiple displacement amplification simulator. In: 2012 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1–4. IEEE (2012)
10. Paez, J.G., Lin, M., Beroukhim, R., Lee, J.C., Zhao, X., Richter, D.J., Gabriel, S., Herman, P., Sasaki, H., Altshuler, D.: Genome coverage and sequence fidelity of Φ29 polymerasee-based multiple strand displacement whole genome amplification. Nucleic Acids Res. **32**(9), e71 (2004)
11. Li, H., Durbin, R.: Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics **25**(14), 1754–1760 (2009)
12. Arriola, E., Lambros, M.B., Jones, C., Dexter, T., Mackay, A., Tan, D.S., Tamber, N., Fenwick, K., Ashworth, A., Dowsett, M.: Evaluation of Phi29-based whole-genome amplification for microarray-based comparative genomic hybridisation. Lab. Invest. **87**(1), 75–83 (2007)
13. Bredel, M., Bredel, C., Juric, D., Kim, Y., Vogel, H., Harsh, G.R., Recht, L.D., Pollack, J.R., Sikic, B.I.: Amplification of whole tumor genomes and gene-by-gene mapping of genomic aberrations from limited sources of fresh-frozen and paraffin-embedded DNA. J. Mol. Diagn. **7**(2), 171–182 (2005)
14. Zhang, C., Zhang, C., Chen, S., Yin, X., Pan, X., Lin, G., Tan, Y., Tan, K., Xu, Z., Hu, P.: A single cell level based method for copy number variation analysis by low coverage massively parallel sequencing. PLoS ONE **8**(1), e54236 (2013)

Joint Tree of Combinatorial Maps

Tao Wang(✉), Congyan Lang, and Songhe Feng

School of Computer and Information Technology, Beijing Jiaotong University,
Beijing 100044, China
{twang, cylang, shfeng}@bjtu.edu.cn

Abstract. Combinatorial maps are widely used in the field of computer vision, including image segmentation, medical image analysis and mobile robotics. Many practical problems can be formulated as the combinatorial map matching problem. This paper addresses the problem of inexact matching between labeled combinatorial maps. We define Joint Tree of combinatorial maps, and prove it can be used to decide of map isomorphism. In this way, the map matching problem is relaxed to the Joint Tree matching problem, which can be solved in polynomial time. Our approach provides a novel way to explore the problem of combinatorial map matching.

Keywords: Combinatorial map · Map matching · Joint tree · Image analysis

1 Introduction

In the healthcare field, images, and especially digital images, are produced in ever-increasing quantities and used for diagnostics and therapy. Thus, image representation, segmentation and retrieval are import issues in this field. Compared with traditional graph model, combinatorial maps are more powerful structures for modeling topological structures with subdivided objects. The concept was first introduced informally for modeling planar graphs [1], and was later extended to represent higher-dimensional subdivided objects. Compared with traditional graph-based representations, combinatorial maps have some natural advantages. First, combinatorial maps are more precise for explicitly encoding the orientations of edges around vertices. Second, it is easy to descript high-dimensional patterns using combinatorial maps. Because of its precision and simplicity, the combinatorial map model has been used in many fields, including image representation [2, 3], 3D medical image analysis [4–6], mobile robotics [7, 8], and mathematical chemistry [9].

Matching combinatorial maps is therefore an important problem in the field of healthcare and image analysis. There have been some previous works on the map matching problem. The early research can be traced back to Cori who discussed the computation of the automorphism group of a topological graph embedding in his report [10]. Liu defined sequence descriptions for combinatorial maps [11], which are subsequently used for map isomorphism and map automorphism [12]. Gosselin et al. proposed two map signatures which are used to efficiently search for a map in a database [13]. Damiand et al. proposed a polynomial algorithm for searching compact submap in planar maps, and then extended this work to n-dimensional open

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 22–28, 2014.
DOI: 10.1007/978-3-319-13186-3_3

combinatorial maps [14]. Wang et al. proposed a quadratic algorithm for submap isomorphism based on sequence searching [15].

All these works described above are only for the exact map matching problem, which aim for finding an exact one-to-one mapping between two combinatorial maps. In real applications, two objects having small structural differences are usually considered as matched. Also, real world objects are usually affected by noises so that map representations extracted from identical objects at different time are rarely exactly equal. Therefore, it is necessary to integrate some degree of error-tolerance into the map matching process. Combier et al. defined the first error-tolerant distance measure for comparing generalized maps by means of the size of a largest common submap [16], and then related maximum common submaps with the map edit distance by introducing special edit cost functions [17]. This approach cannot be directly used for comparing labels on the maps. In most scenarios maps extracted from real world objects are always labeled. Wang et al. defined edit distance of combinatorial maps and proposed an optimal algorithm to compute map edit distance based on tree search [18]. This approach is more flexible in terms of the ability of comparing labels on maps. However, all these approaches have exponential computational complexity and are difficult to be applied in real applications. It therefore demands research attention on exploring more efficient approach on measuring distance between labeled maps.

In this paper, we address the problem of measuring the distance between two combinatorial maps, which is one of the most important and fundamental issues of the inexact map matching problem. In particular, we aim to efficiently solve the labeled map matching problem via relaxing the problem to tree matching. We first extend the concept of *Joint Tree* to combinatorial maps, and propose an efficient algorithm for construction of joint trees. Then we prove that joint trees can be used to decide of map isomorphism, and show how to use joint trees to measure the distance between combinatorial maps. In this way, the map matching problem is relaxed to the tree matching problem, which can be solved in polynomial time.

2 Background

In this section, we recall some basic notions of combinatorial maps. Concepts and terminologies not mentioned here can be found in [18].

A 2D combinatorial map may be understood as a graph explicitly encoding the orientation of edges around a given vertex. The basic element in combinatorial maps is called dart, and each edges is composed of two darts with different direction. The fact that two darts stem from the same edge is recorded in the involution α. A permutation σ defines the rotation of darts around a vertex. Each cycle of σ is associated to one vertex and encodes the orientation of darts encountered when turning counterclockwise around this vertex (e.g. the σ-cycle (3, 4, −1) in Fig. 1).

Definition 1. (2D labeled combinatorial map) A 2D labeled combinatorial map G is a 4-tuple $G = (D, \alpha, \sigma, \mu)$ where

- D is a finite set of darts,
- α is the involution on D,

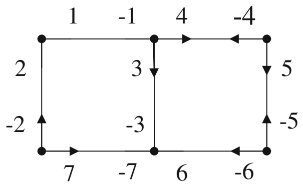


Fig. 1. A combinatorial map G_1

- σ is the permutation on D,
- and μ is a dart label function.

Figure 1 demonstrates the derivation of a combinatorial map from a plane graph, where $D = \{1, -1, 2, -2, 3, -3, 4, -4, 5, -5, 6, -6\}$, $\alpha = (1, -1)(2, -2)(3, -3)(4, -4)(5, -5)(6, -6)(7, -7)$ and $\sigma = (1, 2)(3, 4, -1)(5, -4)(7, -2)(6, -3, -7)(-5, -6)$. Usually, μ is a partial function mapping darts to a finite set of integers, characters or vectors. A labeled map $G = (D, \alpha, \sigma, \mu)$ is connected if for any two darts x and y in D, y can be reached from x by successive applications of the involution α and the permutation σ. For the sake of simplicity, maps in this paper are connected and vertices are unlabeled unless otherwise stated.

Definition 2. (map isomorphism) Given two labeled maps $G_1 = (D_1, \alpha_1, \sigma_1, \mu_1)$ and $G_2 = (D_2, \alpha_2, \sigma_2, \mu_2)$, if there is a one-to-one mapping ψ: $D_1 \rightarrow D_2$ such that for any $x \in D_1$, there are

$$\psi(\alpha_1(x)) = \alpha_2(\psi(x)), \quad \psi(\sigma_1(x)) = \sigma_2(\psi(x)), \quad \mu_1(x) = \mu_2(\psi(x)),$$

then G_1 and G_2 are considered isomorphic.

3 Joint Tree of Combinatorial Maps

3.1 Definition and Construction

Liu introduced joint trees of a graph to solve the embedding distributions of a graph by genus [15]. In this section, we extend joint trees to combinatorial maps and give its construction algorithm.

Definition 3. (joint tree of combinatorial maps) Given a combinatorial map $G = (D, \alpha, \sigma, \mu)$ and a dart $d \in D$, the joint tree of G associated with d, denoted as $JT(G, d)$, is a 3-tripes (T, s, μ) returned by the *JT_Construction* algorithm described in Algorithm 1, where

- T is a rooted ordered tree,
- s is a symbol function mapping each vertex of T to a nonnegative integer,
- and μ is a label function mapping each vertex of T to a pair (l_1, l_2), in which l_1 and l_2 a two labels derived from labels of darts in G.

Definition 4. (betti-vertex) A vertex u in a joint tree is a betti-vertex if $s(u) > 0$.

Definition 5. (partner vertex) Two vertices u and v in a joint tree are two partner vertices if $s(u) = s(v)$ and $s(u) > 0$, and we say u (respectively v) is the partner vertex of v (respectively u) in this case.

Definition 6. (map-to-tree mapping) Given a combinatorial map $G = (D, \alpha, \sigma, \mu)$ and one of its joint tree $JT(G, d) = (T, s, \mu)$ ($d \in D$), the map-to-tree mapping φ: $D \rightarrow T$ is defined in company with the construction of the joint tree as in Algorithm 1.

Algorithm 1. *JT_Construction* (map G, dart d)

Input: a combinatorial map $G=(D, \alpha, \sigma, \mu)$, and a dart $d \in D$.

Output: the joint tree $JT(G, d) = (T, s, \mu)$.

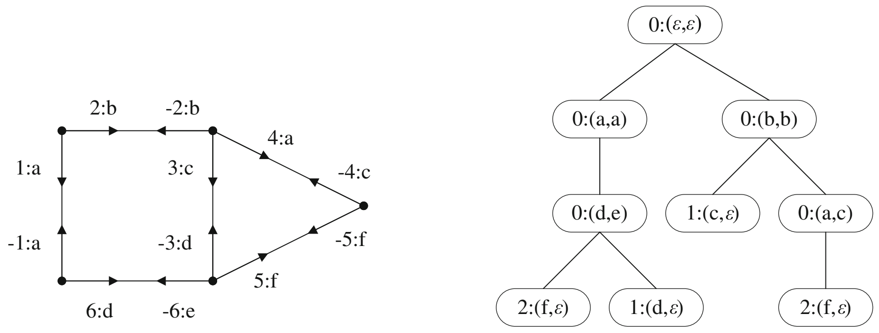
Comments: queue Q stores the temporary darts to be visited.

Comment: h_x and t_x denote head vertex and tail vertex of dart x respectively.

Comment: ε denotes a special empty label.

Comment: m_x denote a temporary mark of dart x.

```
Initialize queue $Q = \Phi$ , $l = 1$,
Push darts $d$ and $\alpha(d)$ into $Q$, and mark vertices $t_d$ and $h_d$ as a visited.
Initialize $T$ containing root vertex $t_d$ and its child $h_d$.
Set $s(t_d) = 0$, $s(h_d) = 0$, $\mu(t_d) = (\varepsilon, \varepsilon)$, $\mu(h_d) = (\mu(d), \mu(\alpha(d)))$.
while $Q$ is not empty do
    pop the first dart, say $x$, from $Q$.
    let $y = \sigma(x)$.
    while $y <> x$ do
        if vertex $h_y$ is not visited then
            push dart $\alpha(y)$ into $Q$, and mark vertex $h_y$ as visited.
            Insert vertex $h_y$ into $T$ as the last child of $t_y$.
            Set $s(h_y) = 0$, $\mu(h_y) = (\mu(y), \mu(\alpha(y)))$, $\varphi(y) = \varphi(\alpha(y)) = h_y$.
        else
            Insert a new vertex $u$ into $T$ as the last child of $t_y$,
            Set $\mu(u) = (\mu(y), \varepsilon)$, $\varphi(y) = u$.
            if dart $y$ is not marked then
                Mark $y$ and $\alpha(y)$ with $m_y = m_{\alpha(y)} = 1$, and set $s(u) = l$.
                Set $l = l + 1$.
            else
                Set $s(u) = m_y$.
            end if
        end if
        Let $y = \sigma(y)$.
    end while
end while
End Algorithm
```



(a) map G_0

(b) joint tree $JT(G_0, 1)$

Fig. 2. A sample of joint tree of a combinatorial map

For illustrating the procedures of Algorithm 1, we construct the joint tree of the sample combinatorial map G_0 with dart 1 (see Fig. 2(a), in which the character of each dart indicates the label of the dart). The final joint tree $JT(G_0, 1)$ are shown in Fig. 2(b), in which the integer and the pair of characters in each vertex are its symbol and labels respectively.

3.2 Decision of Isomorphism

Given rooted ordered tree T, we denote the root of T as $r(T)$. For each vertex u in T, we denote the degree of u as $d(u)$ and the ith child of u ($0 < i \leq d(u)$) as $c(u, i)$.

Definition 7. (Joint tree isomorphism) Given two joint trees $JT_1 = (T_1, s_1, \mu_1)$ and $JT_2 = (T_2, s_2, \mu_2)$, if there is a one-to-one mapping ψ: $T_1 \rightarrow T_2$ such that $\psi(r(T_1)) = r(T_2)$, and for any vertex u in T_1 there are

- $\psi(c(u,\ i)) = c(\psi(u), i)$ for any $0 < i \leq d(u)$,
- $s_1(u) = s_2(\psi(u))$,
- $\mu_1(u) = \mu_2(\psi(u))$,

then JT_1 and JT_2 are considered isomorphic.

The decision of isomorphism between two joint-trees is to check whether there is such a mapping from T_1 to T_2. It can be conducted by a *DFS*(Depth First Search) traversal, and its computational complexity is $O(|T_1|)$.

Lemma 1. Given two maps $G_1 = (D_1, \alpha_1, \sigma_1, \mu_1)$ and $G_2 = (D_2, \alpha_2, \sigma_2, \mu_2)$, and two darts $d_1 \in D_1$ and $d_2 \in D_2$, the two joint trees $JT(G_1, d_1) = (T_1, s_1, \mu_1)$ and $JT(G_2, d_2) = (T_2, s_2, \mu_2)$ are isomorphic ***if*** G_1 and G_2 are isomorphic with d_1 mapping to d_2.

Proof. Let φ_1 be the map-to-tree mapping from G_1 to $JT(G_1, d_1)$, φ_2 be the map-to-tree mapping from G_2 to $JT(G_2, d_2)$, and ψ_1 be the isomorphism mapping between G_1 and G_2 such that $\psi_1(d_1) = d_2$. Obviously, there exist a mapping ψ_2: $T_1 \rightarrow T_2$ such that $\psi_2(r$

$(T_1)) = r(T_2)$ and $\psi_2(\varphi_1(x)) = \varphi_2(\psi_1(x))$ for any $x \in D_1$. Then, for any vertex u in T_1, we have for any

$$\begin{aligned} 0 < i \leq d(u), \psi_2(c(u,i)) &= \psi_2(\varphi_1(\sigma_1^i(\varphi_1(u)))) \\ &= \varphi_2(\psi_1(\sigma_1^i(\varphi_1(u)))) \\ &= \varphi_2(\sigma_2^i(\psi_1(\varphi_1(u)))) \\ &= \varphi_2(\sigma_2^i(\varphi_2(\psi_2(u)))) \\ &= c(\psi_2(u), i) \end{aligned}$$

It is undoubtable that the root of $T(M_1, x_1)$, say r_1, equals to the root of $T(M_2, x_2)$, say r_2. The ith child of r_1 (respectively r_2) is the head vertex of dart $P_1^{i-1}x_1$ (respectively $P_2^{i-1}x_2$). Let ψ be the isomorphic mapping from M_1 to M_2 such that $\psi(x_1) = x_2$, it can be deduced that $P_2^{i-1}x_2 = \psi(P_1^{i-1}x_1)$ according to the Definition 1. So the ith child of r_1 equals to ith child of r_2. It is the same case for any vertex and its children in $T(M_1, x_1)$ and $T(M_2, x_2)$, thus $T(M_1, x_1)$ and $T(M_2, x_2)$ are equivalent.

According to Lemma 1, two strictly matched combinatorial maps have the same joint trees. So, we claim that the similarity of two combinatorial maps depend to a great extend on the similarity of their joint trees. We define the distance between two maps as

$$d(G_1, G_2) = \min\{d(JT(G_1, d_1), JT(G_2, d_2)|d_1 \in D_1, d_2 \in D_2\} \tag{1}$$

In this way, we relax the problem of map distance measure to the problem of tree distance measure, which can be solved in polynomial time [19].

4 Conclusion

As the combinatorial map model has been used in field of computer vision, matching combinatorial maps is therefore an important problem in the field of pattern recognition and image analysis. This paper addresses the problem of inexact matching between labeled combinatorial maps. We define the concept of Joint Tree of combinatorial maps, and prove that it can be used to decide of map isomorphism and to measure the distance between combinatorial maps. In this way, the map matching problem is relaxed to the tree matching problem. Our approach provides a novel way to explore the problem of combinatorial map matching.

Acknowledgements. This work is supported by the National Nature Science Foundation of China (Nos. 61300071, 61272352, 61100142) and the Fundamental Research Funds for the Central Universities (No. 2012JBM040).

References

1. Jones, G.A., Singerman, D.: Theory of maps on orientable surfaces. Proc. Lond. Math. Soc. **3**, 273–307 (1978)
2. Brun, L., Kropatsch, W.: Contraction kernels and combinatorial maps. Pattern Recogn. Lett. **24**(8), 1051–1057 (2003)

3. Brun, L., Kropatsch, W.: Contains and inside relationships within combinatorial pyramids. Pattern Recogn. **39**(4), 515–526 (2006)
4. Damiand, G.: Topological model for 3D image representation: definition and incremental extraction algorithm. Comput. Vis. Image Underst. **109**(3), 260–289 (2008)
5. Dupas, A., Damiand, G.: First Results for 3D image segmentation with topological map. In: Coeurjolly, D., Sivignon, I., Tougne, L., Dupont, F. (eds.) DGCI 2008. LNCS, vol. 4992, pp. 507–518. Springer, Heidelberg (2008)
6. Heimann, T., MeinZer, H.P.: Statistical shape models for 3D medical image segmentation: a review. Med. Image Anal. **13**(4), 543–563 (2009)
7. Dufourd, D., Chatila, R.: Combinatorial maps for simultaneous localization and map building(SLAM). In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 1047–1052 (2004)
8. Teng, Z., Kang, D.J.: Disjunctive normal form of weak classifiers for online learning based object tracking. In: Proceedings of VISAPP, vol. 2, pp. 138–146. SciTePress (2013)
9. Ramon, G.D., Jorge, G., Jesus, J.O., Lionello, P.: Some new trends in chemical graph theory. Chem. Rev. **108**(3), 1127–1169 (2008)
10. Cori, R.: Computation of the automorphism group of a topological graph embedding. Technical report (1985)
11. Liu, Y.P.: Advances in Combinatorial Maps. Northen Jiaotong University Press, Beijing (2003). (In Chinese)
12. Wang, T., Liu, Y.P.: Implements of some new algorithms for combinatorial maps. OR Trans. **12**(2), 58–66 (2008)
13. Gossenlin, S., Damiand, G., Solnon, C.: Efficient search of combinatorial maps using signatures. Theor. Comput. Sci. **412**(15), 1392–1405 (2011)
14. Damiand, G., Solnon, C., Higuera, C., Jandodet, J.-C., Samuel, E.: Polynomial algorithms for subisomorphism of nD open combinatorial maps. Comput. Vis. Image Underst. **1157**, 996–1010 (2011)
15. Wang, T., Dai, G.J., Xu, D.: A polynomial algorithm for submap isomorphism of general maps. Pattern Recogn. Lett. **32**, 1100–1107 (2011)
16. Combier, C., Damiand, G., Solnon, C.: Measuring the distance of generalized maps. In: Jiang, X., Ferrer, M., Torsello, A. (eds.) GbRPR 2011. LNCS, vol. 6658, pp. 82–91. Springer, Heidelberg (2011)
17. Combier, C., Damiand, G., Solnon, C.: From maximum common submaps to edit distances of generalized maps. Pattern Recogn. Lett. **33**(25), 2020–2028 (2012)
18. Wang, T., Dai, G.J., Ni, B., Xu, D., Siewe, F.: A distance measure between labeled combinatorial maps. Comput. Vis. Image Underst. **116**(12), 1168–1177 (2012)
19. Bille, P.: A survey on tree edit distance and related problems. Theor. Comput. Sci. **337**(1), 217–239 (2005)

An Optimization Method of Fusing Multiple Decisions in Object Detection

Zhu Teng(✉) and Baopeng Zhang

School of Computer and Information Technology,
Beijing Jiaotong University, Beijing, China
{zteng, bpzhang}@bjtu.edu.cn

Abstract. Object detection is widely employed in a large number of areas, such as human detection, medical image processing, etc. However, it is insufficient to use only a learning algorithm to detect objects and more techniques or models, such as a probability based approach, a part model, a segmentation model, are combined with the learning algorithm to accomplish the detection task. To this end, a fusion approach is required to balance the decisions making by multiple models. This paper proposes an optimization methodology that fuses a set of confidence outputs estimated by multiple models. Various experiments are executed and demonstrate that the proposed fusion method has a relative better performance than that of the system constituted by a single model.

Keywords: Fusing multiple decisions · Optimization method · Object detection

1 Introduction

Computer vision can be used in assorted areas, such as security (Li and Shen 2013), detection (Pedro F. Felzenszwalb et al. 2010), retrieval (Jun Wu et al. 2013), and so on, among which object detection is one of the most important areas and is widely employed in various applications. For instance, in medical image processing where the detection of anatomical objects such as the heart plays a significant role in assisting the clinicians in diagnosis, therapy planning and image-guided interventions (Yang Wang et al. 2013). The detection task has been studied by many researchers for decades and many successful results have been reported. In general, the detector that finds the bounding boxes of objects in images was realized by some learning algorithm in most works. The learning method used in object detection can be a boosting algorithm (Ivan Laptev 2009), a supported vector machine (SVM) (Scholkopf and Smola 2002), a transformation of any of them (Andreas Opelt et al. 2006) or a combination of some of them (Zheng Song et al. 2011). Besides, probability based approaches (Michael C. Burl et al. 1998) were also involved by many researchers. They are mainly utilized to encode the spatial relationship in a graphical model (Zhu Teng et al. 2014; Justin Domke et al. 2013) such as the Bayesian model (Bogdan Alexe et al. 2010), a pictorial structure (Fischler and Elschlager 1973), a tree model (Long (Leo) Zhu et al. 2010) and so on. Some other object detection methods employed a segmentation model (Bastian Leibe et al. 2008) and others build a hierarchy model to represent the object by layers

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 29–35, 2014.
DOI: 10.1007/978-3-319-13186-3_4

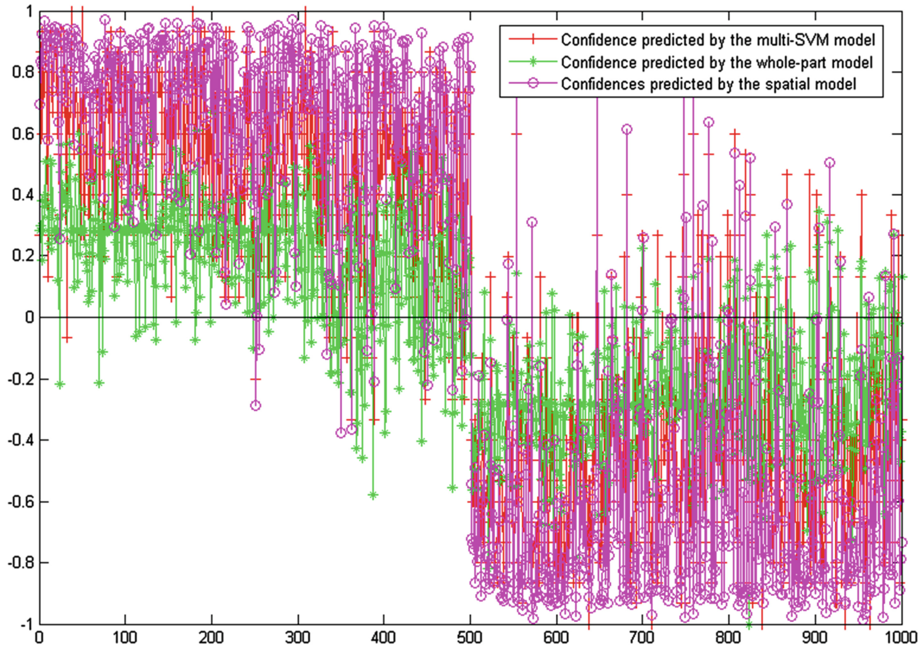


Fig. 1. An example of the confidences estimated by a learning method, a part model, and a probabiltiy based model.

(Zhu and Mumford 2006). It is much easier to detect an object if an accurate segmentation of the test image is obtained. To sum up, it is insufficient to use only a learning algorithm to detect objects and more auxiliary techniques or models (such as a probability based approach, a part model (Pedro F. Felzenszwalb et al. 2010), a graph based model (Tao Wang et al. 2012), a segmentation model, a saliency detection model (C. Lang et al. 2012), etc.) are combined with the learning method to accomplish the detection task. To this end, a fusion approach is required to balance the decisions making by multiple models.

As a motivation example, we assume three types of models, including multi-SVM model (Zhu Teng et al. 2014), part model (Pedro F. Felzenszwalb et al. 2010), spatial relationship model, are employed in the object detection task. The confidences that are estimated by these three models are based on a detection window separately. Figure 1 shows the confidences predicted by the multi-SVM model, part model, and spatial relationship model on 1000 training examples (positive: 1 $\sim$ 500, negative: 501 $\sim$ 1000) of the UIUC Image Database for Car Detection.[1] It can be seen from the figure that there are erroneous predictions for all three models, but the inaccurate confidences (such as points in the left bottom region and the top right region) predicted by these three models are not always on the same example. Therefore, if an elegant

[1] The UIUC Image Database for Car Detection is available at http://cogcomp.cs.illinois.edu/Data/Car/.

combination of these three models can be exploited, a higher accuracy of the object detection algorithm could be achieved. In brief, a fusion approach is necessary to combine these confidences in order to give a wise decision on a detection window.

In this paper, we propose an optimization approach that fuses multiple decisions made by several models to obtain higher accuracy and better performance for object detection. The remains of the paper are arranged as follows. Section 2 describes the optimization approach. Section 3 shows the experiments to demonstrate the institution of the optimization method and we come to the conclusions in Sect. 4.

2 Fusion Approach

In this section, the optimization method to combine multiple decisions is delineated. The multiple decisions are generally represented by confidences ranged from 0 to 1. If they are not, the confidences should be normalized first. Without loss of generality, three decisions making by multi-SVM model, part model and spatial relationship model are assumed and employed in this work. The goal of the fusion of the multi-SVM model, part model and spatial relationship model is to diminish the erroneous decisions making on the detection windows. An objective function is defined and an optimization method is employed to find the minimum of this objective function and the corresponding values of the variables. The optimization problem is formulated as described in Eq. (1).

$$\begin{aligned} \underset{\alpha,\beta,\gamma,\delta}{\text{minimize}} \quad & -\sum_{i=1}^{n} sign(\alpha \cdot C_{m_i} + \beta \cdot C_{p_i} + \gamma \cdot C_{c_i} + \delta) \cdot y_i \\ \textit{subject to} \quad & \alpha + \beta + \gamma + \delta = 1 \end{aligned} \tag{1}$$

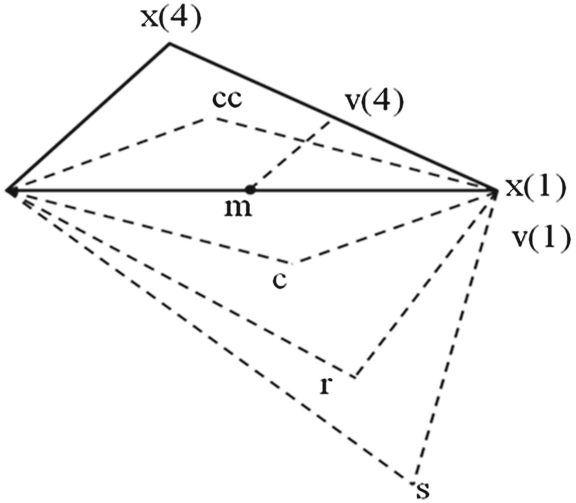


Fig. 2. Nelder-Mead (NM) simplex algorithm. The bold outline is the original simplex and the dashed outline indicates a possible new simplex.

C_{m_i}, C_{p_i}, and C_{c_i} in the objective function of Eq. (1) are the confidences on the i^{th} detection window estimated by the multi-SVM model, part model and spatial relationship model, respectively. y_i is the ground truth of the i^{th} detection window (*-1* suggests a detection window without the object and *1* indicates a detection window with the object). n is denoted as the total number of detection windows. The meaning of the objective function is the minus of the number of correctly estimated detection windows, and to minimize it is to maximize the number of detection windows that are accurately determined. α, β, γ, and δ are the optimization variables.

As the sign function and several variables are involved in the objective function, a multivariable nonlinear optimization method is required to acquire the minimum of the objective function. The method employed in this research is the Nelder-Mead (NM) simplex algorithm (J. A. Nelder, R. Mead 1965; J.C. Lagarias et al. 1998), which is an unconstrained nonlinear optimization method, and the proposed optimization problem is required to be described in an unconstrained form as shown in Eq. (2).

Table 1. NM simplex algorithm.

1) Let $x(i)$ denote the list of points in the current simplex, $i = 1,\ldots,4$.
2) Order the points in the simplex from the lowest objective function value $f(x(1))$ to the highest $f(x(4))$. At each step in the iteration, the algorithm discards the current worst point $x(4)$, and accepts another point into the simplex. Or, in the case of step 7, it changes all *4* points with values above $f(x(1))$.
3) Generate the reflected point $r = 2m - x(4)$, where $m = \sum_{i=1}^{3} x(i)/3$, and calculate $f(r)$.
4) If $f(x(1)) \le f(r) < f(x(3))$, accept r and terminate this iteration.
5) If $f(r) < f(x(1))$, calculate the expansion point $s = m + 2(m - x(4))$ and $f(s)$.
    a. If $f(s) < f(r)$, accept s and terminate the iteration.
    b. Otherwise, accept r and terminate the iteration.
6) If $f(r) \ge f(x(3))$, perform a contraction between m and the better one of $x(4)$ and r:
    a. If $f(r) < f(x(4))$ (i.e., r is better than $x(4)$), calculate $c = m + (r - m)/2$ and $f(c)$. If $f(c) < f(r)$, accept c and terminate the iteration. Contract outside. Otherwise, continue with Step 7).
    b. If $f(r) \ge f(x(4))$, calculate $cc = m + (x(4) - m)/2$ and $f(cc)$. If $f(cc) < f(x(n+1))$, accept cc and terminate the iteration. Contract inside. Otherwise, continue with Step 7).
7) Calculate the three points $v(i) = x(1) + (x(i) - x(1))/2$ and $f(v(i))$, $i = 2,\ldots,4$. The simplex at the next iteration is $x(1)$, $v(2),\ldots,v(4)$.

$$\underset{\alpha,\beta,\gamma}{\text{minimize}} \quad -\sum_{i=1}^{n} sign(\alpha \cdot C_{m_i} + \beta \cdot C_{p_i} + \gamma \cdot C_{c_i} + 1 - \alpha - \beta - \gamma) \cdot y_i \qquad (2)$$

The NM simplex algorithm belongs to the general class of the direct search method that does not utilize any derivative information. It uses a simplex of $t + 1$ points for the t-dimensional vectors $\mathbf{x}$. The algorithm first makes a simplex around the initial guess $\mathbf{x_0}$, and then updates the simplex repeatedly according to the following steps (refer to (J.C. Lagarias et al. 1998) and (J. A. Nelder, R. Mead 1965) for the details of the algorithm, and here only the necessary procedures for the optimization are given, see also Fig. 2). Since there are only three parameters in the proposed formulation, t is three in this case. The objective function is denoted by $f(x)$ ($\mathbf{x}$ denotes the variables α, β, γ) for short. Table 1 gives the details of the algorithm.

Since the NM algorithm starts at an initial estimate and finds a local solution, the NM algorithm is proceeding at different initial values a thousand times in order to avoid falling into a local optimum.

3 Experiments

In this section, the performance of the fusion method is reported on the test datasets of two categories, airplane and car, and the program is coded by Matlab. The validation dataset is distinct from both the training dataset and test dataset.

The UIUC Image Database for Car Detection contains training images, single-scale test images, and multi-scale test images, and the validation dataset is built by the single-scale test images and the test dataset is constructed by the multi-scale test images. The Caltech Airplanes dataset[2] dataset consists 1074 images and is divided into a training set (500 images), a validation set (74 images) and a test set (500 images). The three models are examined on the validation dataset, and the confidences that the multi-SVM model, part model and spatial model estimate on the images are extracted and utilized to learn optimization parameters α, β, γ. Note that there might be more than one detection window for an image from the validation dataset. The label (-1 or 1) of a detection window is following the criterion of PASCAL (Mark Everingham et al. 2010), which is obtained by comparing the detection window with the annotation (bounding box) of the corresponding image. If the overlap between the detection window and the ground-truth bounding box of the image exceeds 50 %, the detection window is considered as true. Multiple detections of the same object are considered false. For example, *4* detections of a single object (the overlap of all *4* detections is over 50 %) in an image should be counted as *1* correct detection and *3* false detections.

Table 2 presents the results on the test dataset. The performance of this experiment is evaluated by the percentage of the number of correctly detected windows to the total number of windows. A detection window is considered as true if the confidence is positive; otherwise, it is regarded as false. The only multi-SVM model of Table 2 means that only the multi-SVM confidence are used to make decisions on the examples

[2] The Caltech Airplanes dataset is available at http://www.vision.caltech.edu/html-files/archive.html.

Table 2. Accuracy comparison of the fusion method on the test dataset.

Category	Only multi-SVM model	Only part model	Only spatial model	Fusion method
Airplane	0.3155	0.4660	0.4078	0.6845
Car	0.4683	0.5000	0.6901	0.8028

in this model, and so as the only part model and the only spatial model. The confidence of the fusion method is a combination of these three confidences as discussed in Sect. 2 (α, β, γ are determined by the NM simplex algorithm). It is clear from Table 2 that the fusion method outperforms any of the other models for each category and it could further improve the performance of the object detection system.

4 Conclusions

As the accuracy of object detection is demanded higher and higher, detection by only a learning algorithm is insufficient and multiple models are entailed to be combined. In this paper, we propose an optimization method to combine multiple decisions making by a learning method and some other models or techniques. The experiments on the benchmark datasets demonstrate that the fusion method performs better than any single model that composes the fusion approach.

Acknowledgements. This work was supported by the Fundamental Research Funds for the Central Universities with grant number 2014JBM040 and Natural Science Foundation of China (61370070).

References

Opelt, A., Pinz, A., Zisserman, A.: Incremental learning of object detectors using a visual shape alphabet. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 3–10 (2006)

Leibe, B., Leonardis, A., Schiele, B.: Robust Object Detection with Interleaved Categorization and Segmentation. Int J Comput Vis **77**, 259–289 (2008). doi:10.1007/s11263-007-0095-3

Scholkopf, B., Smola, A.J.: Learning with Kernels, Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning). The MIT Press, Cambridge (2002)

Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2010)

Lang, C., Liu, G., Yu, J., Yan, S.: Saliency detection by multitask sparsity pursuit. IEEE Trans. Image Process. **21**(3), 1327–1338 (2012)

Laptev, I.: Improving object detection with boosted histograms. Image Vis. Comput. **27**, 535–544 (2009)

Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**, 308–313 (1965). doi:10.1093/comjnl/7.4.308

Lagarias, J.C., Reeds, J.A., Wright, M.H., Wright, P.E.: Convergence properties of the Nelder-Mead simplex method in low dimensions. SIAM J. Optim. **9**(1), 112–147 (1998)

Wu, J., Shen, H., Li, Y.-D., Xiao, Z.-B., Ming-Yu, L., Wang, C.-L.: Learning a hybrid similarity measure for image retrieval. Pattern Recogn. **46**(11), 2927–2939 (2013)

Domke, J.: Learning graphical model parameters with approximate marginal inference. PAMI, 35 (10), pp. 2454–2467 (2013) (to appear)

Zhu, L., Chen, Y., Yuille, A., Freeman, W.: Latent hierarchical structural learning for object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2010)

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**, 303–338 (2010). doi:10.1007/s11263-009-0275-4

Fischler, M.A., Elschlager, R.A.: The representation and matching of pictorial structures. IEEE Trans. Comput. **c-22**(1), 67–92 (1973)

Burl, M.C., Weber, M., Perona, P.: A probabilistic approach to object recognition using local photometry and global geometry. In: Proceedings of European Conference on Computer Vision, pp. 628–641 (1998)

Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. 32(9), 1627–1645 (2010)

Zhu, S.-C., Mumford, D.: A stochastic grammar of images. Found. Trends Comput. Graph. Vis. **2**(4), 259–362 (2006). doi:10.1561/0600000018

Wang, T., Dai, G., Ni, B., Xu, D., Siewe, F.: A distance measure between labeled combinatorial maps. Comput. Vis. Image Underst. **116**(2012), 1168–1177 (2012)

Wang, Y., Georgescu, B., Chen, T., Wen, W., Wang, P., Xiaoguang, L., Lonasec, R., Zheng, Y., Comaniciu, D.: Learning-based detection and tracking in medical imaging: a probabilistic approach. Lect. Notes Comput. Vis. Biomech. **7**, 209–235 (2013)

Li, Y., Shen, H.: On identity disclosure control for hypergraph-based data publishing. IEEE Trans. Inf. Forensics Secur. **8**(8), 1384–1396 (2013)

Song, Z., Chen, Q., Huang, Z., Hua, Y., Yan, S.: Contextualizing object detection and classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011)

Teng, Z., Zhang, B., Kim, O., Kang, D.-J.: Regional SVM classifiers with a spatial model for object detection. In: International Conference on Computer Vision Theory and Applications, Lisbon, Portugal (2014)

Data Mining and Decision Analytics for Public Health and Wellness

Mining Typical Order Sequences from EHR for Building Clinical Pathways

Shoji Hirano(✉) and Shusaku Tsumoto

Department of Medical Informatics, School of Medicine,
Shimane University, 89-1 Enya-cho, Izumo, Shimane, Japan
hirano@ieee.org, tsumoto@computer.org

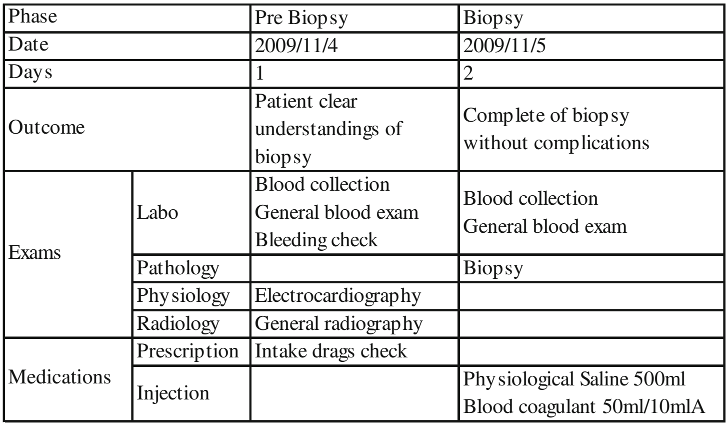
Abstract. Clinical pathway is one of the key tools for providing standardized treatment for patients. However, building a new pathway from scratch is a time-consuming task for medical staffs, as it involves optimization of the treatment plan while preserving operability in a hospital. In this paper, we present a method for mining typical treatment processes from electric health records (EHRs) for facilitating creation of new pathways by providing base blocks. Firstly, we constitute occurrence and transition frequency matrices of clinical orders using all cases. Next, we compute the typicalness index for each order sequence based on the occurrence and transition frequencies. After that we perform clustering of all cases according to the similarity defined on the typicalness index. Experimental results on two disease datasets demonstrate that the method is capable of producing clusters that reflect differences of treatment processes without a priori information about order types.

1 Introduction

Clinical pathways (also called care pathways, care maps etc.) [1] receive much attention in hospitals as useful tools for managing treatment plans. A medical team (doctors, nurses, and other co-medicals) cooperatively builds up a template that contains the common treatment plan for a disease, considered as the best according to the available evidences, and share it among the team. Figure 1 shows an example of clinical pathways. In this example, the columns correspond to time and represent treatment phases such as pre-operation day, operation day (biopsy in this case) and post operation day. The rows are tasks such as examinations, treatments and prescriptions that should be done on the corresponding phases. Expected outcomes may also be defined on each phase in order to assess whether the treatment goes well and the patient is ready for discharge or for the next phase of treatment.

When building a new pathway, it is recommended to ensure that the pathway contains a treatment plan reasonably arranged for early discharge and is easily adaptable to the current work-flow in the hospital, although it can be further improved by successive revisions after launch. Creating such a pathway from scratch imposes very hard work to medical staffs. Therefore, in practice,

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 39–49, 2014.
DOI: 10.1007/978-3-319-13186-3_5



Phase		Pre Biopsy	Biopsy
Date		2009/11/4	2009/11/5
Days		1	2
Outcome		Patient clear understandings of biopsy	Complete of biopsy without complications
Exams	Labo	Blood collection General blood exam Bleeding check	Blood collection General blood exam
	Pathology		Biopsy
	Physiology	Electrocardiography	
	Radiology	General radiography	
Medications	Prescription	Intake drags check	
	Injection		Physiological Saline 500ml Blood coagulant 50ml/10mlA

Fig. 1. An example of clinical pathways.

some doctors firstly pick up several cases in which patients could smoothly discharge from the hospital, and then refine their treatment plans with other staffs to make a template. However, even if we take such an approach, selecting representative cases from history databases is still time-consuming as it involves comparison of treatment processes among patients. By providing a tool for supporting this task, it may be possible to facilitate the creation of new clinical pathways.

In this paper we attempt to extract typical treatment processes that were applied to a large portion of patients by comparing clinical order histories stored electronically on a hospital information systems. Even when a standard treatment plan is available for a disease, it is rather rare to see that practically issued orders (such as examinations, prescriptions) were completely the same among the patients. This may be due to the differences on patient conditions, on the versions of plans applied, date or time of operations, calender dates, and so on. Our task is to find core treatment processes constituted of widely applied sets of orders taking into account these factors.

2 Method

2.1 Overview

We firstly determine a target disease for which we want to make a pathway, and prepare a dataset for analysis by extracting relevant clinical orders from the hospital information systems. Next, we create occurrence and transition frequency maps using all the cases (i.e. all the order sequences) in the dataset. Then we compute the 'typicalness' of each order sequence according to the generated maps. After that, we perform clustering of the order sequences based on the similarity we define on the typicalness. Finally, we select from each cluster one sequence with the highest typicalness value as a candidate of the core

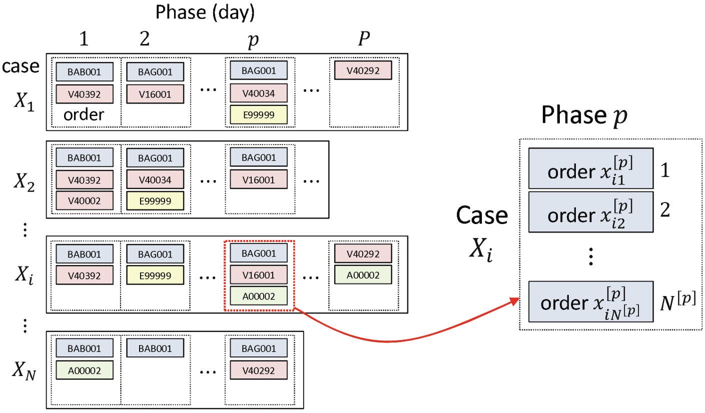


Fig. 2. Data structure.

process. The use of occurrence and transition frequency maps enables us to take into account the importance of orders in terms of their application ratio when comparing order sequences. By clustering, we try to discover various types of treatment processes reflecting the differences on strategies or conditions for treatment. In the following subsections we describe data structure and the detail of each process.

2.2 Data Structure

Figure 2 shows the data structure used in the proposed method. By N we denote the number of cases in the dataset. By $X_1, X_2, \ldots, X_N$ we denote each case, where a case corresponds to one admission of a patient to the hospital. Each case contains clinical orders issued by medical staffs. By $p (p = 1, 2, \ldots, P)$ we denote a phase, which corresponds to the days elapsed from the admission. The number of phase, P, would be different among cases depending on the length of stay in the hospital. Note that a phase can be defined arbitrary; we employed one-day as a phase for simplicity, but it is also possible to employ another interval, e.g. a half day, as a phase. We gather orders issued on the same phase into a set, and represent them as a sequence of order sets. By x we denote each order. An order issued for case X_i at the m-th in phase p is denoted by $x_{im}^{[p]}$. The number of orders issued in a phase, denoted by $N^{[p]}$, can be different for each phase and each case.

An order is categorized by the corresponding order code described by an alphabet and digits. For example, the code V40392 represents a nursing order "check for dietary intake", and the code A00002_N50449 represents a prescription order "Isodine 7 % 30 ml for gargling". Note that these codes are dependent on the hospital information systems we used and other systems may employ other codes; however it does not directly affect the methodology proposed in this paper.

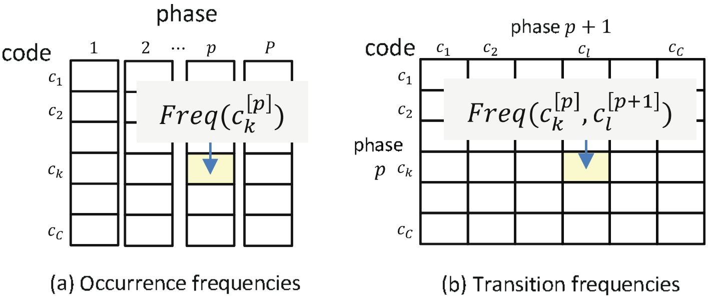


Fig. 3. Occurrence and transition matrices.

2.3 Occurrence and Transition Frequency Maps

We firstly count occurrence frequencies of orders at each phase using all data in the dataset. We define $\mathrm{Freq}(c_k^{[p]})$ that represents the frequency of issuing an order with code c_k at phase p over all cases as follows.

$$\mathrm{Freq}(c_k^{[p]}) = \left\| \left\{ x_{im}^{[p]} \in U | \mathrm{code}(x_{im}^{[p]}) = c_k \right\} \right\| \tag{1}$$

where $U = \{X_1, X_2, \ldots, X_N\}$ denotes the set of all cases and $\mathrm{code}(x_{im}^{[p]})$ does the order code for order $x_{im}^{[p]}$. The symbol $\| \cdot \|$ denotes the cardinality of a set.

Next we count frequencies of order transitions between two adjacent phases. Let us denote by $\mathrm{Freq}(c_k^{[p]}, c_l^{[p+1]})$ the frequency of observing two orders $c_k^{[p]}$ and $c_l^{[p+1]}$ at phase p and $p+1$ respectively on the same case. We define the transition (co-occurrence) frequency as follows.

$$\begin{aligned}&\mathrm{Freq}(c_k^{[p]}, c_l^{[p+1]}) = \\ &\left\| \left\{ (x_{im}^{[p]}, x_{in}^{[p+1]}) \in U | (\mathrm{code}(x_{im}^{[p]}) = c_k) \wedge (\mathrm{code}(x_{in}^{[p+1]}) = c_l) \right\} \right\|\end{aligned} \tag{2}$$

The occurrence and transition frequencies can be stored in matrices as shown in Fig. 3. Assuming that the dataset contains C types of orders, the occurrence frequencies are stored in a $C \times P$ matrix whose element corresponds to $\mathrm{Freq}(c_k^{[p]})$ as show in Fig. 3(a). As for the transition frequencies, a $C \times C$ matrix whose element corresponds to $\mathrm{Freq}(c_k^{[p]}, c_l^{[p+1]})$ is used for a pair of adjacent phases $(p, p+1)$. A total of $P-1$ matrices are prepared for storing transition frequencies where P denotes the number of phases. Note that we add to the set of order codes $\{c_1, c_2, \cdots, c_C\}$ an arbitrary code representing existence of no order.

2.4 Typicalness Index

Based on the occurrence and transition frequency maps generated in the previous section, we derive the typicalness of each order sequence. We evaluate the

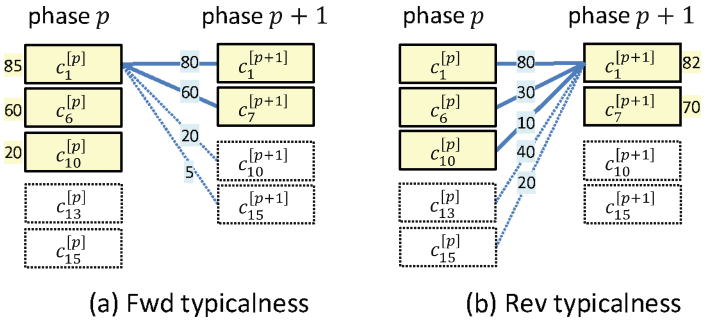


Fig. 4. Forward and reverse typicalness indices.

typicalness of a sequence bi-directionally in the forward direction looking from phase p to $p+1$ and in the reverse direction from phase $p+1$ to p. Figure 4 provides an illustrative example. In Fig. 4, an order surrounded by a solid rectangle denotes that the order is included in the target case we are evaluating its typicalness. An order surrounded by a dotted rectangle denotes that the order is not included in this target case (i.e., included in other cases in the dataset). The value put beside each order represents the occurrence frequency of that order in the whole dataset. The value put on the line between two orders represents the transition (co-occurrence) frequency of these orders. Each of them respectively corresponds to $\text{Freq}(c_k^{[p]})$ and $\text{Freq}(c_k^{[p]}, c_l^{[p+1]})$ introduced in the previous section.

In Fig. 4(a), the target case contains three types of orders in phase p, that are, $c_1^{[p]}$, $c_6^{[p]}$, $c_{10}^{[p]}$. The three values put on the left of these orders, 85, 60 and 20, respectively indicate the total number of cases in the dataset that also have these orders in phase p. Orders such as $c_{13}^{[p]}$ and $c_{15}^{[p]}$ exist in the dataset but are not included in the target case that we are currently evaluating; thus represented by dotted rectangles.

Now we focus on the transition from $c_1^{[p]}$. In the whole dataset, we observe that four types of orders, $c_1^{[p+1]}$, $c_7^{[p+1]}$, $c_{10}^{[p+1]}$ and $c_{15}^{[p+1]}$ could be issued after $c_1^{[p]}$ in phase $p+1$. Their frequencies are 80, 60, 20, and 5 as shown on the corresponding lines. Among the four, the target case contains $c_1^{[p+1]}$ and $c_7^{[p+1]}$ (surrounded with a solid rectangle) whose transition frequencies are 80 and 60. As these transition frequencies are relatively large and those for the remaining two orders are small (20 and 5), we can infer that this target case includes order transitions that are widely observed in the dataset, i.e., it includes typical patterns of order transitions.

Based on these values, we make an index representing how much of the widely applied order transitions are included in the target case. We firstly compute the sum of transition frequencies for orders present in the target case ($80+60 = 140$) and divide it by the sum of frequencies over all orders ($80 + 60 + 20 + 5 = 165$). Next we put a weight on it using the occurrence frequency (=85) of the preceding order $c_1^{[p]}$. The resultant value constitutes a forward typicalness index

$(85 \times 140/165 = 72.1)$ about order $c_1^{[p]}$ in this target case. By applying the same process to the other two orders in phase p and taking sum of them, we obtain the forward typicalness index of the target case on phase $p \rightarrow p+1$.

We formalize the forward typicalness index as follows.

$$\begin{aligned}\mathrm{FwdTypicalness}^{[p]}(X_i) = &\sum_{m=1}^{N^{[p]}} Freq(c_m^{[p]}) \\ &\times \frac{\sum_{n=1}^{N^{[p+1]}} Freq(c_m^{[p]}, c_n^{[p+1]})}{\sum_{l=1}^{C} Freq(c_m^{[p]}, c_l^{[p+1]})}\end{aligned} \tag{3}$$

For simplicity, we here denote by $c_m^{[p]}$ the order code of $x_{im}^{[p]}$, originally denoted by $\mathrm{code}(x_{im}^{[p]})$. The forward typicalness index takes low value when the target case does not have in phase $p+1$ the subsequent orders exhibiting high transition frequencies.

Based on the same scheme we also define the reverse typicalness index as follows.

$$\begin{aligned}\mathrm{RevTypicalness}^{[p]}(X_i) = &\sum_{n=1}^{N^{[p+1]}} Freq(c_n^{[p+1]}) \\ &\times \frac{\sum_{m=1}^{N^{[p]}} Freq(c_m^{[p]}, c_n^{[p+1]})}{\sum_{k=1}^{C} Freq(c_k^{[p]}, c_n^{[p+1]})}\end{aligned} \tag{4}$$

Figure 4(b) provides an example. RevTypicalness takes low value when preceding orders with high transition frequencies are not included in phase p on the target case.

We then sum up the forward and reverse typicalness indices and average them by the number of orders included in each phase. The resultant value is the typicalness index of case X_i in phase p. It is formalized as follows.

$$\mathrm{Typicalness}^{[p]}(X_i) = \frac{\mathrm{FwdTypicalness}^{[p]}(X_i) + \mathrm{RevTypicalness}^{[p]}(X_i)}{N^{[p]} + N^{[p+1]}} \tag{5}$$

The final typicalness of case X_i is obtained by summing up the typicalness values and number of orders over all phases and taking the average as follows.

$$\mathrm{Typicalness}(X_i) = \frac{\sum_{p=1}^{P}(\mathrm{FwdTypicalness}^{[p]}(X_i) + \mathrm{RevTypicalness}^{[p]}(X_i))}{\sum_{p=1}^{P}(N^{[p]} + N^{[p+1]})} \tag{6}$$

2.5 Clustering Based on the Typicalness Index

Based on the typicalness index, we perform clustering of the cases in the dataset. We compare the indices phase-by-phase, not as a whole, so that the similarity on the distribution of the indices over phases is captured. We define the similarity between cases X_i and X_j, $\mathrm{Dissim}(X_i, X_j)$, as follows.

$$\mathrm{Dissim}(X_i, X_j) = \sum_{p=1}^{P} \left(\mathrm{Typicalness}^{[p]}(X_i) - \mathrm{Typicalness}^{[p]}(X_j) \right)^2 \tag{7}$$

When the numbers of phases are different on cases X_i and X_j, the typicalness index for absent phase is treated as zero. For example, when X_i is constituted of 8 phases and X_j is constituted of 9 phases, the dissimilarity on phase 9 is derived by assuming the typicalness of X_i is zero on phase 9.

3 Experimental Results

We generated two datasets for experiments by extracting clinical orders from our hospital information systems. The first dataset is associated with an otolaryngological disease, and the second dataset is associated with an obstetric disease. All cases in a dataset had the same DPC (diagnosis procedure combination) code. We selected these diseases because we already have manually created clinical pathways in use and hence they were suitable for examining the basic ability of forming groups of typical treatment processes from the mixture data.

We computed the typicalness index for each case and the dissimilarity for each pair of cases by using the proposed method. Then we performed hierarchical clustering of the cases by using Ward's method [2].

3.1 Results on the Otolaryngological Disease Dataset

For this disease we had three kinds of pathways dealing with extraction of a certain region A. In this paper we respectively denote them by (1) Region A extraction, (2) Region A extraction (AM OP) and (3) Region A extraction (PM OP), where AM OP means surgery in the morning and PM OP means surgery in the afternoon. Historically, (1) was firstly created and it was later branched into (2) and (3) so that care processes could be optimized for AM or PM operations. Each of the three pathways experienced revisions within the data collection period of 2.5 years; there were two versions for (1), five versions for (2) and similarly five versions for (3). A total of 158 cases were included in the dataset. Among them, 78 were applied either one of the clinical pathways and the remaining 80 were not applied any pathways.

Figure 5 shows the dendrogram generated by Ward's method. The dendrogram revealed that there were two or three large clusters in the dataset. We started investigating the characteristics of clusters in a bottom-up manner from seven clusters solution, where the dissimilarity stepped up largely at an early phase of cluster formation. For further references, we put cluster indices 1–7 on the dendrogram.

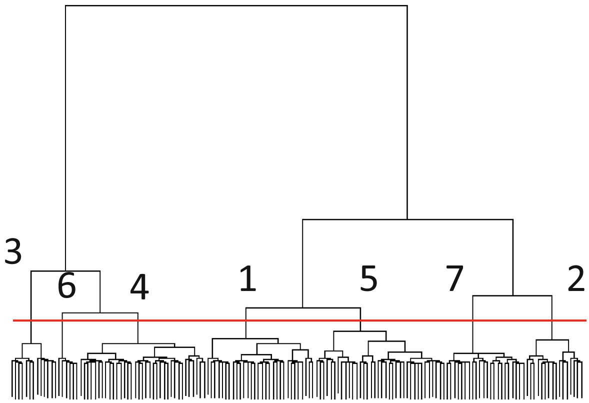


Fig. 5. Dendrogram for the otolaryngological disease dataset.

Table 1. Cluster statistics for the otolaryngological disease dataset.

Cluster index	Number of cases	Phase of operation (Mean ± SD)	Typicalness index (Mean ± SD)
3	13	3.9 ± 0.3	17.7 ± 3.8
6	6	-	6.9 ± 5.6
4	35	4.0 ± 0.7	12.5 ± 3.5
1	30	2.0 ± 0.0	30.9 ± 4.5
5	36	2.0 ± 0.2	24.1 ± 4.9
7	22	2.2 ± 0.6	21.9 ± 3.6
2	16	2.0 ± 0.0	23.6 ± 6.1
Total	158	-	21.3 ± 8.1

Tables 1 and 2 provide the following statics for each cluster: (a) mean and standard deviation of the phase of operation (corresponds to the elapsed days from admission to the operation), (b) mean and SD of the typicalness index values, (c) the number of cases for each type of pathways, and (d) the ratio of applying pathways. The order of clusters in the table corresponds to the order of cluster indices put on the dendrogram.

The ratios of applying clinical pathways were low (<0.3) in clusters 3, 6 and 4. Especially, cluster 6 was purely made of cases in which no clinical pathways were applied. The average phases of operations in these clusters located around the fourth day. We could observe clear differences between these clusters and others such as cluster 1, whose average phase of operations located around the second day. Cluster 6 exhibited the lowest typicalness index, and other two clusters 3 and 4 similarity exhibited low typicalness values. These observations suggested that the first (top) branch of the dendrogram reflected differences of treatment processes associated with the differences of the phase of operation. For cluster 6,

Table 2. Cluster statistics for the otolaryngological disease dataset (continued).

Cluster index	Number of cases for each type of pathway					Ratio of applying pathways
	Region A extraction			No pathways applied	Total	
	-	AM OP	PM OP			
3	1	0	2	10	13	0.23
6	0	0	0	6	6	0.00
4	5	3	2	25	35	0.29
1	4	10	14	2	30	0.93
5	6	4	9	17	36	0.53
7	2	3	2	15	22	0.32
2	1	7	3	5	16	0.69
Total	19	27	32	80	158	0.49

we did not compute the average phase of operations because we could not find operation order in four cases out of six; but the remaining two had operations on the fourth and seventh days respectively.

Clusters 1, 5, 7 and 2 that located on the right branch of the dendrogram exhibited converse characteristics. Their average phases of operations located around the second day and their typicalness indices were relatively high. Especially, the ratio of applying clinical pathways in cluster 1 was quite high (>0.9). Taking this highest typicalness value into account, we considered that this cluster was made of the cases in which standardized care processes were applied based on clinical pathways. The group of four clusters (1, 2, 5, 7) was made from the two subgroups of clusters (1, 5) and (2, 7) according to the dendrogram. From the number of cases stratified by the name of pathways shown in Table 1, we could observe that in clusters 1 and 5 the ratio of applying the pathway 'Region A extraction (PM OP)' was higher than other pathways, while in clusters 7 and 2 the ratio of applying the pathway 'Region A extraction (AM OP)' was higher than other pathways. Cluster 5 included much larger number of cases with no pathways compared to cluster 1. The similar characteristics could be observed for cluster 7. Fisher's test on the four clusters (1, 2, 5, 7) versus four pathways (including no pathway applied) yielded $p < 0.001$, which implied statistical dependence between these two factors. Therefore, we consider that the proposed method could capture the similarity of care processes related to the phase or time of operations without giving any a priori information about operations.

3.2 Results on the Obstetric Disease Dataset

For this disease we had two kinds of pathways, one for multiparas and another for primiparas. A total of 124 cases were included in the dataset. Among them,

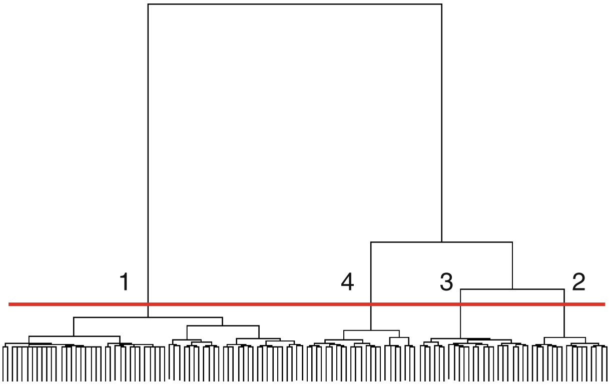


Fig. 6. Dendrogram for obstetric disease dataset.

Table 3. Cluster statistics for the obstetric disease dataset.

Cluster index	Number of cases	Typicalness index (Mean ± SD)
1	62	5.2 ± 2.1
4	23	13.1 ± 2.1
3	23	7.9 ± 2.2
2	16	12.1 ± 2.3
Total	124	8.0 ± 4.0

25 were applied either one of the clinical pathways and the remaining 99 were not applied any pathways. Basically this disease did not involve operations.

Figure 6 shows the dendrogram for obstetric disease dataset produced by Ward's method. Similarly to the otolaryngological disease dataset, we selected the four-cluster solution shown with the horizontal line on the dendrogram for investigation. Tables 3 and 4 provide statistics for generated clusters. Clusters 4 and 2 yielded high average typicalness values, while the other two clusters including the largest one yielded low typicalness values. We could observe clear difference on the distribution of pathways among clusters. Clusters 1 and 2 were composed almost purely of the cases without any pathways (cluster 1) and with the pathway for multiparas (cluster 2) respectively. Cases with the pathway for primiparas were mostly grouped in cluster 4. Fisher's test on the four clusters versus three pathways (including no pathway applied) yielded $p < 0.001$, which implied statistical dependence between these two factors. These results suggested that the proposed method could discover small but relatively dense groups of similar processes in data.

Table 4. Cluster statistics for the obstetric disease dataset (continued).

Cluster index	Number of cases for each type of pathway				Ratio of applying pathways
	For multiparas	For primiparas	No pathways applied	Total	
1	1	0	61	62	0.02
4	0	9	14	23	0.39
3	0	1	22	23	0.04
2	14	0	2	16	0.88
Total	15	10	99	124	0.20

4 Conclusions

In this paper we have presented an attempt for finding clinical pathway candidates based on the typicalness index and cluster analysis. We could observe on the two medical datasets remarkable dependency between clusters and types of pathways. The proposed method computed typicalness index for each order sequence and performed clustering without any a priori information about order type such as operation; therefore, it could be considered that the typicalness measure contributed in quantifying differences of treatment processes. It remains as a future work to evaluate usefulness of the method in creating new clinical pathways.

Acknowledgment. This work was supported in part by the grant-in-aid for scientific research (C) #23500179, by the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

1. Zander, K., Bower, K.: Nursing Case Management, Blueprint for Transformation. New England Medical Center, Boston (1987)
2. Everitt, B.S., Landau, S., Leese, M.: Cluster Analysis, 4th edn. Arnold Publishers, London (2001)

Aspect-Based Sentiment Analysis of Amazon Reviews for Fitness Tracking Devices

Alaa Shafaee(✉), Hassan Issa, Stefan Agne, Stephan Baumann, and Andreas Dengel

German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
alaa.shafaee92@gmail.com

Abstract. The year 2012 marked the birth of a new class of wireless wearable fitness trackers (e.g., Fitbit One) that track daily activity from the count of steps taken and calories burned to stairs climbed and sleep patterns. As the recent trend in research extends the use of these devices to a broader range of applications, questioning the reliability and accuracy of these devices became much more legitimate. In this research, we assess the public opinion on these devices through utilizing novel sentiment analysis techniques to build a fully automated aspect-based sentiment summarizer that transfers the sheer amount of Amazon reviews of these products to a user-friendly summary. Product features are extracted using the text of reviews, the description and features sections on Amazon. Another approach is also proposed that extracts the names of competing products and compares their reviews to separate the features from the other common nouns. To enhance sentiment classication, the system combines two sentiment lexicons, handles complex negation types through parsing while handling semantic relations, and assigns the sentiment tothe proper product and feature. The proposed summarizer's components generally outperform the state-of-the-art methods with notable improvements in detecting product features, competing products and negation and can easily generalize to other domains.

1 Introduction

Fitness tracking has taken a huge leap recently with the introduction of several always-on wearable devices that provide all-time activity tracking. In addition to estimating the number of steps taken, distance traveled and calories burned, the new trackers (e.g., Nike+ Fuelband [3], and Fitbit One [1]) provide easy syncing to smartphones and web portals, third party app development, social sharing and friends competitions. New sensors are continuously incorporated to enable various capabilities including: quality of sleep tracking, stairs climbed counting, heart rate monitoring and skin temperature as well as the recognition of hundreds of different exercises as claimed by the AMIIGO [12] fitness sensor. A recent survey [4], showed that around 5 % of US households with broadband Internet

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 50–61, 2014.
DOI: 10.1007/978-3-319-13186-3_6

own at least one digital fitness device such as the Fitbit trackers. A report [5], showed that 29 % of consumers having health problems would consider using an easy-to-use fitness tracker. Juniper Research [2] forecasts that the number of users of app-enabled health and mobile-fitness hardware devices will increase to 96 millions by 2018, just to name few reports.

Besides using fitness trackers to monitor body wellness, such advances also motivated some doctors to use them to monitor the progress of patients undergoing a surgery or physical therapy [11]. Home automation systems can use sleep tracking to control the lights when users sleep or wake up. Advertisers can target their advertisements to users based on their learned lifestyle. Generally, these devices can help build huge data collections which are very useful to researchers in many disciplines such as marketing, psychology and health care.

In this research, we assess the accuracy and reliability of these devices by building a sentiment summarizer to transfer the sheer volume of Amazon reviews on certain product to a user-friendly summary of people's opinion on each feature of the product. Compared to accessing all available devices through experimental tests, this approach can include all available products and can provide up-to-date information which is based on the experience of many customers for relatively longer time. Despite being optimized to work for fitness tracking devices, many of the methods used in this work can be utilized in any sentiment analysis system. Opinion mining attracts a lot of research efforts due to its numerous challenges. That is, many factors may alter the sentiment of words such as the part-of-speech (POS), position of word in sentence, and presence of negation or sarcasm, context and sentence type like suggestions, conditionals and questions (e.g., *Which* device is the *best?*). Sentiment analysis on aspect level is even more challenging because it requires extracting the features of the product and associating them with the proper sentiment. In addition, extracting the names of competing products dynamically, to associate the sentiment with the correct product, is not frequently addressed in the scope of sentiment analysis. In this work, we tried to overcome many of the aforementioned challenges.

The main contributions of this work are:

- The first application of sentiment summarization in fitness trackers domain which helps customers, manufacturers and researchers interested in using these sensors or the output data of the summarizer with generality to other domains.
- Novel approaches to extract the products features and to handle complex types of negation. Also, novel approaches to dynamically detect the names of competing products in text to associate the sentiment with the correct product and provide users with comparisons between competing products. This also improves the accuracy of features detection.
- To our knowledge, this is the first research in sentiment summarization that utilizes some components offered by Amazon such as the description section of products to make the system components independent of the volume and source of reviews. The same Amazon tools can be used with other sources of reviews or feeds such as tweets. Also, to our knowledge, extracting users recommendations for the final summary is not offered before.

The related work is presented in Sect. 2. The main components of the system are presented in Sects. 3.1–3.5. Next, an evaluation of the main components of the system is offered in Sect. 4.2. Last, Sect. 5 offers the conclusion and future work.

2 Related Work

Sentiment Summarization on aspect level is studied in [8,10,14,17]. The aspects are commonly extracted using frequent nouns and phrases in reviews while applying some filters such as removing stop words and redundant one-term features. It is noticed that most source of error comes from the presence of domain-related nouns that occur frequently such as "weight" in fitness trackers domain. In addition, relying on frequent nouns ignores rare product features. This is handled by our system through utilizing the Amazon APIs. Also, many papers neglect mentions of competing products which affects the results as mentioned in Sect. 3.1.

3 Approach

Stanford NLP tools are used in tokenization, part-of-speech tagging [19], parsing [18], describing the grammatical relationships in sentences [9] and lemmatization. The Product Name Extractor aims at identifying any mention of competing products in text (e.g., Fitbit One and Jawbone Up are competing products). The Aspects Extractor identifies the features of the product (e.g., price and battery life). The Sentiment Extractor classifies the sentiment in a piece of text to positive, negative or neutral. After the sentiment is aggregated with the proper aspect, the data is summarized with statistical and textual proof of people's opinion on product features.

3.1 Extracting Product Names

Given the name of a product, the system uses the Amazon ItemSearch API to get the list of items returned on searching for this product. The product name is then extracted from the product title (e.g., "Fitbit Zip" from "Fitbit Zip - Wireless Activity Tracker"). Notably, product names tend to follow two patterns in terms of the manufacturer name which is also returned by Amazon API: ([manufacturer name] [model name]) as in "Jawbone UP" and ([model name] "by" [manufacturer name]) such as "Up by Jawbone". It is important to note that the data on Amazon can be used to extract the competing products in different opinion sources such as tweets. Part of the product names are commonly used to refer to it such as "the One" instead of "Fitbit One". The most common pattern noticed is ("the" [model name]) with the model name starting in an upper case letter ("The *one* who answered my call is helpful" versus "The *One* is way accurate"). Two other patterns and the names of manufacturers who have only one product in the list of competing products are also matched.

3.2 Aspect Extraction and Domain Modeling

In this section, two approaches are proposed to extract dynamic aspects of products which are product features that may differ from a product to another (e.g., sleep tracker, steps counter and syncing) as opposed to static aspects (e.g., price).

Amazon Tools and Reviews' Text: In this approach, aspects are considered to be elements that occur in both of the following sets.

Set A: Includes the nouns and compound nouns that occur in the structured "product features" section by Amazon and the nouns in all product descriptions provided by the different product sellers on Amazon.

Set B: Includes nouns that frequently co-occur with specific verbs within 10 words range in Amazon's features section (e.g., tracks steps, computes calories). Additionally, the nouns occurring in some syntactic patterns like some of those in [8,14] are included in this set.

Domain-Related Words Filter: In this approach, a filter containing domain-related words is built. First, the intersection of nouns that appear in the reviews of the two products with the highest number of reviews under the class of interest are taken. Second, the resulting list of nouns is manually checked to keep only the very general words. In case of fitness trackers class, the number of common nouns was 1577 which resulted in 1554 after a quick manual refinement of the list.

To extract the dynamic aspects, a list of the lemmas of the nouns in reviews is first collected and sorted by frequency. Lemmatization is important to combine words like "step" and "steps" together. Afterwards, the stop words and any product name are removed from the list. Last, the filter of domain-related nouns generated above is applied and the list is split at certain threshold determined experimentally to separate the potentially candidate aspects from the rest of the list.

Grouping Similar Aspects: Given the name of a core aspect, the words in "synonyms", "co-ordinate terms", "derivations" and "hyponyms" synsets in WordNet [16] are considered different names of the same aspect based on their frequency score. For fitness trackers, a domain model was built based on this approach covering the common aspects (e.g., the price category) with manual refinement.

3.3 Sentiment Extraction

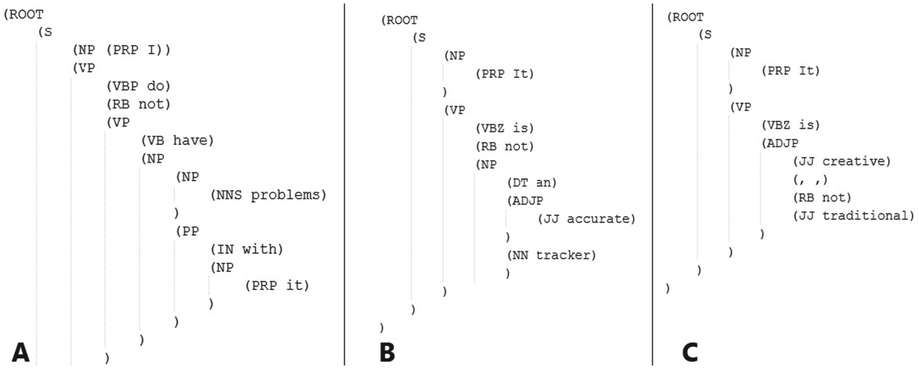
Sentiment Lexicon: A sentiment lexicon is a dictionary of positive and negative words such as "good", and "terrible". SentiWordNet [7] and Liu's Opinion Lexicon [13], are combined in this work to assign a sentiment score to each word in the sentence to take the advantages of both lexicons. SentiWordNet assigns probabilistic sentiment scores to each synset in WordNet, includes 38,182 sentiment-bearing words, and takes the POS into account. However, it is not reviewed by humans. Liu's opinion lexicon is manually complied, includes many misspelled words that appear frequently in social media context, morphological variants, slang, and social-media mark-up.

SentiWordNet assigns a score to each meaning of the word. The meaning of the word is considered to be the most positive (or negative) meaning if it is classified as positive (or negative) by Liu's lexicon. The score computed using SentiWordNet is normalized to get a score between -3 and 3 (inclusive). This score is considered to be the score of the word if both lexicons have the same classification. Otherwise, the sentiment is taken from SentiWordNet if the exact word occurs with a strong sentiment that matches polarity of Liu's lexicon in different part-of-speech. Otherwise, it is considered to be 1 if the polarity assigned by Liu's lexicon is positive and -1 if it is negative. For adjectives and adverbs that do not occur in Liu's lexicon, the score is computed using SentiWordNet by averaging the polarity of synsets under this part-of-speech. A version was slightly modified to include words related to fitness trackers such as "overestimates" or "overcounts" in "It overcounts the number of steps I took".

Handling Negation: Negation words other than "not" and "never" are handled using the window approach [10] which is simple and fast. That is, the polarity of the first 5 words (chosen experimentally) after a negation word is inverted. For example, the negation word "no" in "no help from the customer service" reverses the sentiment score of {help, from, the, customer, service} resulting in changing the polarity of "help" from positive to negative.

The more frequent negation words "not" and "never" are handled by parsing to detect the scope of the negation word better and to handle the semantic relations between phrases in sentence. This handles negation in more complex sentences such as "The man did not succeed in stealing my precious wallet" where "not succeed" and "stealing" are negative, however, the overall sentiment is positive. The three main steps of the algorithm are described below.

1. **Assigning sentiment score to tokens** using our combined lexicon and the components discussed in the following section.
2. **Detecting negation scope** with the help of the Stanford typed dependency tool, which extracts exactly one word negated by each negation word among other relations. For the previous sentence, it returns neg(not, have) among other relations. If the negated word detected by the Stanford tool is a sentiment-bearing word, its polarity is inverted. Otherwise, the scope of negation is detected by the following approach.
   (a) Identify the word negated by each negation word using the typed dependency tool.
   (b) Tag the part-of-speech (POS) of the negated word.
   (c) The nearest verb/noun/adjective phrase is marked if the POS of the negated word is verb/noun/adjective respectively. That is, if the POS of the negated word is a verb as in the previous sentence, the sentence



```
(ROOT
  (S
    (NP (PRP I))
    (VP
      (VBP do)
      (RB not)
      (VP
        (VB have)
        (NP
          (NP
            (NNS problems)
          )
          (PP
            (IN with)
            (NP
              (PRP it)
            )
          )
        )
      )
```

```
(ROOT
  (S
    (NP
      (PRP It)
    )
    (VP
      (VBZ is)
      (RB not)
      (NP
        (DT an)
        (ADJP
          (JJ accurate)
        )
        (NN tracker)
      )
    )
  )
)
```

```
(ROOT
  (S
    (NP
      (PRP It)
    )
    (VP
      (VBZ is)
      (ADJP
        (JJ creative)
        (, ,)
        (RB not)
        (JJ traditional)
      )
    )
  )
)
```

Fig. 1. Parse trees of the referenced examples

marked is the nearest parent/ancestor verb phrase. Thus, in the previous example, the phrase "have problems with it" is marked.

Figure 1B shows the parse tree of "It is not an accurate tracker". The typed dependency tool associates the negation with the noun *tracker*. The marked sentence is the nearest noun phrase, i.e., "an accurate tracker". This leads to inverting the sentiment of *accurate* as per the next step.

(d) If the negation word is outside the marked phrase, the sentiment of all words of the marked phrase is inverted. Otherwise, only the sentiment of words following the negation word in the parse tree is inverted. In Fig. 1A, the sentiment of "have problems with it" is inverted because the negation word is outside the marked phrase which results in changing the sentiment of the whole sentence from negative to positive.

The parse tree of "It is creative, not traditional" is shown in Fig. 1C. The negation is associated with the adjective "traditional". The nearest adjective phrase is "creative, not traditional". Since the negation word is inside the adjective phrase, only the part following it (i.e., traditional) gets inverted. Thus, the sentiment of *creative* is not inverted.

3. **Handling semantic relations:** This is done similar to [6]. The polarity of the parent nodes of the leaves of the tree (i.e., words) is the sum of the sentiment score of its children. The polarity of the rest of the tree is computed recursively as follows. The polarity of the deepest rightmost level of the tree with the level just above it are used to determine their overall polarity intuitively according to Table 1. As shown in Fig. 2, the right most level is "my precious wallet" whose polarity is determined to be positive from the previous step. The higher level "stealing" is negative. This means doing something negative to something positive is negative as the third entry of the table indicates. The part "stealing my precious bag" is replaced accordingly with a negative node and the process is recursively repeated till only one node is left which represents the overall sentiment of the tree. Similarly, "have not succeeded" is negative and "breaking my precious bag" is negative which results

in an overall positive sentiment since failing in doing something bad is good as in the last entry of the table. Hence, the overall sentiment of the sentence is positive.

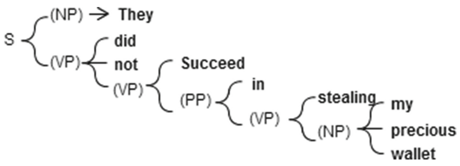


Fig. 2. Parse tree of a negated sentence

Table 1. Semantic relations

Predecessor	Successor	Result
positive	positive	positive
positive	negative	negative
negative	positive	negative
negative	negative	positive

3.4 Handling Other Challenges

Several components were used to refine the accuracy of sentiment classification. A component is used to collect verbs that express functionality as mentioned in Sect. 3.2 and is incorporated with the negation component. Although these verbs are normally neutral, their negation is negative because the device is not functioning properly (e.g., The data collected does not “sync” to my phone). Another component uses the surrounding words of some context-dependent words to detect its sentiment (e.g., lose the “device” versus lose “weight”). Also, The effect of sentiment words is ignored if they come within certain distance from verbs that express wishes and suggestions (e.g., I *wish* they had a *caring* customer service). A list of sentiment diminishers and intensifiers was also used to enhance the sentiment calculation (e.g., It *barely* works). The product name extractor component is used to make sure sentiment is assigned to the correct products.

Comparative Sentences and Users Recommendations: Comparative sentences (e.g., The Fitbit One is more accurate in steps than the Jawbone Up) are extracted by identifying mentions of competing products as in Sect. 3.1. Our system distinguishably offers recommendations of consumers in the final summary. Identifying consumers recommendations is done by matching patterns that frequently carry user suggestions and recommendations. One such pattern is “It would be [positive adjective] to have”.

Assigning Sentiments to Aspects is based on the following assumptions. The sentiment of a sentence mentioning exactly one aspect represents the aspect. The sentiment of each aspect in sentences with multiple aspects is considered to be the sentiment of the deepest parse tree that contains only that aspect. Sentences that do not contain any aspects while mentioning the device name or the word “product” or “device” are considered to describe the whole product. Otherwise, sentiment bearing sentences are considered to describe the last aspect mentioned.

3.5 Summary

The user is presented an easy-to-digest visual summary of people's opinion on different product aspects. The number and percentage of positive and negative reviews about each aspect is presented with a reference to the sentences indicating so. A comparison of people's opinion on different products is also offered. Our system uniquely presents the users' recommendations. Users can also view the trend of people's opinion on different products and their aspects per time. People's testimonials on competing products are also presented.

4 Evaluation

4.1 Data Set

The data used is a set of the Amazon reviews of three leading fitness trackers, namely, Fitbit One, Jawbone Up and Nike+ Fuelband until June, 2013. The star rating, title, date, product ID and author ID are also extracted. Human annotators were asked to annotate 106 reviews with a total of 908 sentences that are almost equally divided between the three products. Reviews were collected from the five different star ratings with equal proportions. Out of the 738 sentences annotated, 280 were purely negative, 301 are purely positive, 123 are neutral and 34 carried mixed sentiment.[1]

4.2 Evaluation of Major System Components

Product Name Extraction: Out of the 43 products returned for Fitbit One, 26 represented different competing products since some results represent the same product by different sellers or in different color. Out of the 26 products, the names of 21 were correctly identified while 1 was not correctly identified. 4 products did not have model name such as "Withings Smart Blood Pressure Monitor" so a wrong word was taken to represent the model. Some of the results are "Fitbit Flex", "Fitbit Zip", "Withings Pulse", "JAWBONE UP", "Polar H7"and "Nike+ FuelBand". The list of competing products returned for the other trackers was very similar.

This approach outperforms other approaches we considered like extracting the named entities which could not decide if the named entity is a competing product or not (e.g., iPhone or Samsung) and did not tag the names of several competing products. The Amazon Similarity API was also considered but most of the returned results were the same product by different sellers or in different colors. Also, we tried fetching the products under the same product category on Amazon, but the categories are too generic so they do not differentiate between different classes of products.

[1] Some annotators skipped some sentences in some reviews.

Table 2. Aspects detected by the system using the domain filter approach

Product	Aspects
Fitbit One	(step, step count), calorie, (stair, floor), (app, application, iphone app), (website, fitbit website, dashboard, web site), (sleep, sleep tracker, sleep patterns, sleep tracking, sleep mode), friend, pocket, (customer service, customer support), badge, weight loss, activity level, point, fitness pal, distance, battery life, flower, food intake
Jawbone Up	(app, iphone app, up app), (sleep, sleep tracking, sleep mode, sleep tracker, sleep monitor, sleep patterns), step, (alarm, alarm clock), (customer service,jawbone support, jawbone customer, customer support), friend, calorie, up band, battery life, website, (headphone jack, jack), point, food intake, food tracking, replacement band, (power nap, nap), heart rate, expectation, distance, pocket
Nike+ Fuelband	calorie, step, (app, android app, iphone app),(point, fuel, nike fuel, fuel points, fuelpoint), (website, web site, nike website), friend, (customer service, customer support), clasp, screw, heart rate, nike store, rust, battery life, streak, couch, gp, water proof, stair, activity level, ice, nike support, algorithm, distance, lifting, measure, led lights

Aspect Extraction: The very common state-of-the-art approach, which considers the aspects to be the frequent nouns, is used as our baseline. The most common nouns in Fitbit One, Jawbone UP and Nike Fuelband reviews after removing the stop words and ignoring the case respectively are: {fitbit, day, steps, sleep, device, activity, time, calories, weight, stairs, product, clip, sync, food, website, app, thing, data, tracker, computer, days, ultra, track, week}, {band, jawbone, sleep, app, product, time, day, days, steps, device, fitbit, data, sync, food, activity, customer, months, tracking},{nike, band, fuel, fuelband, day, product, calories, days, time, device, points, steps, activity, goal, app, thing, wrist, watch, people, week}. It is noticed that the most common nouns contain many non-aspects and miss core aspects. For Fitbit One, a core aspect came after 175 items because Fitbit has a big number of reviews which increases the number of frequent nouns that are not aspects. Extracting aspects is sensitive to both precision and recall. Clearly, missing important aspects hides important details from the customer. In addition, classifying a word falsely as an aspect affects the sentiment associated with the other aspects in the sentence. The product names like "Ultra", which are classified here as aspects, are pruned in our approach using the product extractor component.

The results of extracting dynamic aspects by the domain filter approach proposed in Subsect. 3.2 are shown in Table 2. The aspects in the table are optionally combined with few static aspects (e.g., price) to form the full list of aspects. The aspects in brackets are those grouped under the same category as presented in Sect. 3.2. The elements in the table represent the lemmas of the aspects. The original nouns of these lemmas are used when searching for the aspects in text (e.g., *GPS* is used to search for *gp*). As shown in Table 2, this approach produces very promising results.

(a) All core aspects are covered in the results, i.e., the recall is 100 % for the three products.
(b) The precision of the method is very high for Fitbit One and Jawbone Up. It decreased with Nike+ Fuelband because the frequency of the first word is much less than that of the most frequent term in the other two products because of reviews for Nike are less in number. This frequency was used to calculate the threshold which leads to covering all words that occur above low frequency (here 5). That's why increasing the threshold in this case increases the precision of the method significantly as can be seen by neglecting the last 2–3 lines of Nike Fuelband aspects in the table.

Sentiment Extraction: Supervised methods used to identify the sentiment per sentence often outperform unsupervised approaches but they have poor ability to generalize to other domains and require a large set of annotated data, storage and often running time. In [20], Support Vector Machine (SVM) trained on unigrams alone outperformed the other features and classifiers considered. SVM[2] and Naive Bayes classifiers trained on unigrams are used in the baseline in the evaluation. These methods were implemented using Rapidminer framework [15]. The first two rows of Table 3 show their results on employing three classes (positive, negative and neutral). The third class in rows 3 and 4 represent the neutral sentences as well as sentences containing mixed opinion. SVM experiments employing 3 classes were based on one-vs-all approach.

Analysis of Sentiment Extraction Results: As shown in Table 3, the best accuracy reached by the supervised approaches was 55.10 % by SVM on employing 3 classes when the third class represent the neutral examples and examples with mixed opinion. Rows 5 in Table 3 represents the accuracy of the proposed system. The accuracy of the system when tested on the annotated data was 67.28 % which outperforms the supervised approaches with 2 and 3 classes. The much bigger advantage is that the system is lexicon-based which makes it possible to generalize to different domains.

The system has a very good recall for the positive class and precision for the negative class relative to the supervised approaches. Some sentences carrying negative opinion do not contain sentiment words which results in low precision for the neutral class such as: {"The person who designed this should not be there any more.", "I will never order from this vendor again"}. Notably, most of this type of sentences carry negative sentiment which decreases the recall for the negative class. This suggested using the review rating to detect the sentiment in sentences that do not contain sentiment words. The polarity of such sentences in reviews with (1 or 2) and (4 or 5) star rating was considered to be negative and positive respectively. The results of this step are shown in row 6. The overall accuracy was increased by 1.84 %.

Some sentences were misclassified as positive such as "There is an awesome comparison of the UP vs. the FitBit Flex here". This error was commonly noticed in conditionals "If it continued to work, it would have been a big motivation to lose weight!". Handling conditionals and questions is expected to increase the precision of the positive class. Handling semantic relations (e.g., "I find it difficult to name flaws.") is expected to increase the accuracy. It was not handled on purpose though since it requires parsing each sentence in each review which is very expensive performance-wise. Also, some annotations were annotated by only one person. The sentiment in some cases can be subjective. Thus, the system may be penalized for a sentence while its result is also acceptable. Ten people were involved in the annotation to make the annotations more representable of the performance of the system. Other less frequent reasons for error occur too, such as misspelled words and sarcasm. As shown in this section,

[2] Based on mySVM implementation by Stefan Rueping.

Table 3. Recall, precision and accuracy of sentiment detection

Classifier	+ve rec.	−ve rec.	Class 3 rec.	+ve pres.	−ve pres.	Class 3 pres.	Accuracy
NB	30.90	25.00	82.11	73.23	46.36	23.71	37.50 %
SVM	44.85	35.59	46.34	58.70	44.25	22.89	41.43 %
NB	23.59	47.06	28.57	76.34	4.35	51.95	27.15 %
SVM	50.17	65.84	11.43	60.40	55.06	12.90	55.10 %
System	80.66	46.61	70.83	76.63	36.79	85.06	65.44 %
System 2	**89.86**	**58.36**	**30.25**	**70.15**	**78.84**	**33.96**	**67.28 %**

the approaches presented in this paper are generally very promising. The system was built in java.

5 Conclusions and Future Work

This work presented an example of research meeting the real world where an aspect-based sentiment summarizer was built to summarize people's opinion in reviews of fitness trackers. According to our results, utilizing Amazon data to extract features and competing products is promising and do not depend much on the volume of reviews. Also, our approach in handling negation enhances the accuracy of sentiment classification.

In our future work, we plan to refine the presented techniques for grouping aspects into categories. We also plan to add a component handling conditional sentences and questions. Detecting similarity between sentences will be useful in decreasing the number of sentences in the final summary. Last, more focused studies in sentiment analysis (e.g., handling questions in sentiment analysis) are needed.

References

1. Fitbit.com. Fitbit One (2014). http://www.fitbit.com/one. Accessed 1 February 2014
2. Mobile Health & Fitness Monitoring, App-enabled Devices & Cost Savings 2013–2018. http://www.juniperresearch.com/reports/mobile_health_fitness. Accessed 10 November 2013
3. Nike+ Fuelband SE. Activity Tracker & Fitness Monitor. http://www.nike.com/us/en_us/c/nikeplus-fuelband. Accessed 1 March 2014
4. Parks Associates Market Focus - Digitally Fit: Healthy Living and Connected Devices. http://www.parksassociates.com/marketfocus/dh-1q-2013. Accessed 1 March 2014
5. Parks Associates report - Health Entertainment: Bringing the Fun to Wellness and Fitness. http://www.parksassociates.com/report/health-entertainment. Accessed 1 March 2014

6. Asmi, A., Ishaya, T.: Negation identification and calculation in sentiment analysis. In: The Second International Conference on Advances in Information Mining and Management, pp. 1–7 (2012)
7. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In: Proceedings of LREC (2010)
8. Blair-goldensohn, S., Neylon, T., Hannan, H., Reis, G., Mcdonald, R., Reynar, J.: Building a sentiment summarizer for local service reviews. In: NLP in the Information Explosion Era (2008)
9. de Marneffe, M., MacCartney, B., Manning. C.: Generating typed dependency parses from phrase structure parses. In: Proceedings of International Conference on Language Resources and Evaluation, pp. 449–454 (2006)
10. Hu, M., Liu., B.: Mining and summarizing customer reviews. In: Proceedings of ACM SIGKDD, pp. 168–177 (2004)
11. Husten, L.: Fitbit Could Help Monitor Progress After Heart Surgery (2013). http://www.forbes.com/sites/larryhusten/2013/08/29/fitbit-could-help-monitor-progress-after-heart-surgery/. Accessed 21 November 2013
12. Amiigo — Explore. http://www.amiigo.co/. Accessed 21 November 2013
13. Liu, B.: Opinion Mining, Sentiment Analysis, Opinion Extraction (2013). http://www.cs.uic.edu/liub/FBS/sentiment-analysis.html. Accessed 30 August 2013
14. Popescu A., Etzioni. O.: Extracting product features and opinions from reviews, pp. 339–346 (2005)
15. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: Yale: Rapid prototyping for complex data mining tasks. In: Proceedings of ACM SIGKDD, pp. 935–940 (2006)
16. Miller, G.: Wordnet: a lexical database for english. Commun. ACM **38**(11), 3941 (1995)
17. Ng., R., Pauls., A.: Multi-document summarization of evaluative text. In: EACL 06: Proceedings of ACL (2006)
18. Socher, R., Bauer, J., Manning, C.D., Ng, A.Y.: Parsing with compositional vector grammars (2013)
19. Toutanova, k., Klein, D., Manning, C., Singer., Y: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of NAACL 03, pp. 173–180 (2003)
20. Pang, B., Lee, L., Vaithyanathan., S: Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of EMNLP (2002)

Biologically Inspired Techniques for Data Mining

An Efficient Drug-Target Interaction Mining Algorithm in Heterogeneous Biological Networks

Congcong Li[1(✉)], Jing Sun[1], Yun Xiong[1], and Guangyong Zheng[2]

[1] Shanghai Key Laboratory of Data Science, School of Computer Science,
Fudan University, Shanghai 200433, People's Republic of China
{congcongli12, jingsun, yunx}@fudan.edu.cn
[2] CAS-MPG Partner Institute for Computational Biology,
Shanghai Institutes for Biological Sciences,
Chinese Academy of Sciences, Shanghai 200031,
People's Republic of China
zhenggy@sibs.ac.cn

Abstract. The identification of interactions between drugs and targets is a key area in drug research. Exploring targets can help identify potential side effects and toxicities for drugs, as well as new applications of existing drugs. Because of the enormous scale of biological dataset, most of the existing algorithms for drug-target mining are time-consuming. In this paper, we proposed an optimization algorithm called LSH-HeteSim to mine the drug-target interaction in heterogeneous biological networks, where the relationship between drugs and targets is various. It means drugs and targets are connected with complicated semantic path. In practice, the similarity measure used for semantic path is a path-dependent method, called HeteSim, which had been utilized in some previous studies of relevance search. Experiment results in real biological networks show that our algorithm can effectively predict drug-target interaction with the *AUC* measure achieving 0.943. Simultaneously, the running time of our algorithm is much less than the state-of-art methods.

Keywords: Drug target · Link prediction · Heterogeneous biological networks · Meta-path · Similarity measure

1 Introduction

The key issue of modern drug development is to recognize drug targets. Drug targets are binding sites of drug and biological macromolecules regulated by the drug, such as receptors, enzymes, ion channels, transporters, genes and the like [1]. As the basis of drug discovery and designing, prediction of drug-target interactions is an important issue in the field of biological research field [2].

In traditional biological studies, whether there is a link between the drug and target gene is inferred from biology experiment results which have a long and costly experimental period [3]. In recent years, many endeavors have been made in drug-target interaction prediction, for accelerating drug targets finding, shortening research

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 65–76, 2014.
DOI: 10.1007/978-3-319-13186-3_7

cycle, and reducing development costs [4]. However, most of the endeavors only considered partly characteristics of drug biological networks, such as drug chemical characteristics, or local link information. A biological network is a complex heterogeneous network, which either contains multiple types of objects or multiple types of links [7, 8]. Commonly, in a biological network, the relationship between drug and target is heterogeneous [9, 11], where two objects connected via different paths have different meaning [10]. Hence, for drug target similarity measurement, the semantic paths must be taken into account. Besides, the dataset of biological network is general large, and similarity search algorithms frequently cost a long time. To solve the problem mentioned above, we propose a new drug-target interaction mining algorithm, for similarity path search in heterogeneous biological networks. The new algorithm is named LSH-HeteSim, which is based on the locality sensitive hash method (LSH) [17]. The HeteSim similarity measure method has been utilized in some relevance search problem of social network [10], and it is also employed in our study. Experiments results show that the algorithm we proposed can predict missing links between drugs and targets and identify drug-target interaction with a fairly high accuracy (the *AUC* measure achieve 0.943). Specially, the running time of our algorithm is much less than the state-of-art methods.

The rest of this paper is organized as follows. In Sect. 2, we review some related work about link prediction and existing predictors for drug-target interaction finding. Next, in Sect. 3, we provide detail information of the relevance search algorithm HeteSim and our new drug-target interaction mining algorithm LSH-HeteSim. Then in Sect. 4, we perform two groups of experiments to prove the effectiveness and efficiency of our algorithm. Finally, in Sect. 5, we give discussion and conclusion of this study.

2 Related Work

2.1 Link Prediction

Link prediction aims at estimating the likelihood of the existence of a link between two nodes [12]. In some networks, especially biological networks such as PPI, metabolic networks and food webs, the discovery of links is costly in the laboratory [13]. Instead of blindly checking all possible links, predictions based on the observed links and focusing on those links which are most likely to exist can sharply reduce the experimental costs assuming that the prediction algorithm is accurate.

Node similarity based link prediction method can be roughly categorized into two types: feature based approaches and link based approaches. The feature based approaches measure the similarity of nodes based on their feature values, such as cosine similarity, Jaccard coefficient and Euclidean distance. The link based approaches measure the similarity of nodes based on their link structures in a network. The similarity measure (HeteSim) used in our proposed LSH-HeteSim algorithm is a link based method, especially, it takes the semantic paths into account.

2.2 Methods for Drug-Target Interaction Prediction

The prediction of drug-target interaction is an important research problem in the drug discovery filed. Traditional methods for drug target prediction are based on biological experiments. Due to the long test period and its high cost, there have been more and more drug target prediction methods by calculating. For example, Campillos and Monica proposed a method for identifying drug target genes by the similarity of side effects [4]. He Zhisong and Zhang Jian proposed a method based on the drug and biological characteristics of the functional group [5]. Chen Bin and Ying Ding used a statistical model called SLAP to measure the relationship between the drug and target gene in a biological network [6, 18]. As SLAP takes the heterogeneity of biological networks and the impacts of different semantic paths to the drug targets' similarities into consideration during its statistical computation, it achieves good prediction accuracy. However, due to its complicate computation, the experiments are conducted with sub-datasets on small scale. Once it applies to large scale dataset, the query task will be very time-consuming. Besides considering the semantic paths in heterogeneous biological networks, the optimized algorithm proposed in paper, called LSH-HeteSim, also adopt the LSH. It can reduce the running time sharply without losing the prediction accuracy. We will compare the prediction accuracy of these two algorithms on the same dataset in experimental section.

3 Drug-Target Interaction Mining in Heterogeneous Biological Networks

3.1 Heterogeneous Biological Networks

A heterogeneous network is a special type of network which either contains multiple types of objects or multiple types of links [10], while a heterogeneous biological network is composed of multiple biological objects. In many applications, the network object is no longer constituted with a simple type, but includes various types of objects and links, such as the gene regulatory networks. Here, we give a definition of heterogeneous biological networks as shown in Definition 1.

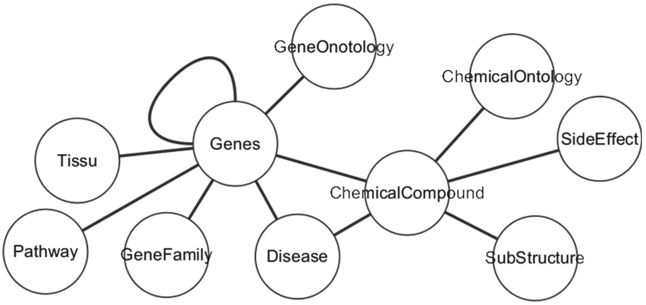


Fig. 1. Network schema of Slap

Definition 1. Heterogeneous biological Networks

Given a network schema $S_G = (A, R)$ which consists of a set of biological objects types $A = \{A\}$ and a set of relations $R = \{R\}$, an biological network is defined as a directed graph $G = (V, E)$ with an object type mapping function $\emptyset(v) \in A$ and a link type mapping function $\varphi(e) \in R$. Each object $v \in V$ belongs to one particular object type $\emptyset(v) \in A$, and each link $e \in E$ belongs to a particular relation $\varphi(e) \in R$. When the types of objects $|A| > 1$ or the types of relations $|R| > 1$, the network is called heterogeneous biological network. Otherwise it is a homogeneous biological network.

Figure 1 is the network schema of a heterogeneous biological network Slap [6]. Each circle represents a biological object type, such as drug, disease, gene, and etc. Each horizontal line represents a relation type, such as the treatment between drug and disease.

Definition 2. Meta-Path

A meta-path P is a path defined on the graph of network schema $T_G = (A, R)$, and is denoted in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \ldots \xrightarrow{R_l} A_{l+1}$, which defines a composite relation $R = R_1 \circ R_2 \ldots \circ R_l$ between types A_i and $A_{l\ +\ 1}$, where $\circ$ denotes the composition operator on relations.

Different from those in homogeneous network, the paths in heterogeneous network called meta-path have semantics, as Definition 2 defined, which make the relatedness between two objects different on different search paths. Taking the heterogeneous network in Fig. 2 for example, the relationship between the drug and the disease is treatment and being treated, and between genes and disease is the causing and caused by. Obviously, drug C_1 is not related to gene G_2 based on CDG path. It means that drugs can treat diseases caused by genes. However, drug C_1 is related to gene G_2 based on CDCDG path because of drug C_2 which can also treat disease D_2.

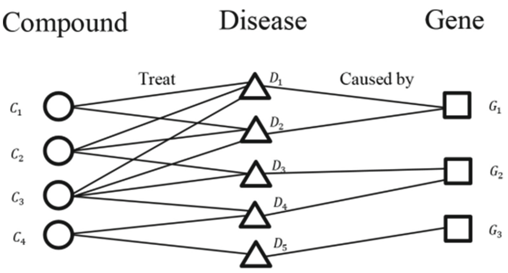


Fig. 2. A simple heterogeneous network example

3.2 Relevance Search

Shi Chuan and Kong Xiangnan have defined the relevance search problem in heterogeneous networks. They proposed an algorithm based meta-path a meta-path called HeteSim to measure the similarity between objects of different types [10]. In this paper, HeteSim was used to assess the interactions between drugs and targets in heterogeneous networks.

Definition 3. HeteSim [10]

$$HeteSim(s,t|R_1 \circ R_2 \circ \ldots \circ R_l) = \frac{1}{|O(s|R_1)||I(t|R_l)|} \sum\nolimits_{i=1}^{|O(s|R_1)|} \sum\nolimits_{j=1}^{|I(t|R_l)|} HeteSim(O_i(s|R_1), I_j(t|R_l)|R_2 \circ \ldots \circ R_{l-1}) \tag{1}$$

Given a meta-path $P = R_1 \circ R_2 \ldots \circ R_l$, HeteSim($s$, $t|P$) is the similarity between object s and t. Here, $O(s|R_1)$ is the out-neighbors of s based on the path P, and $I(s|R_l)$ is the in-neighbors of t based on P. The result it returns is a similarity value between 0 and 1. The larger HeteSim value is, the more similar the two objects are. The similarity of drug C_1 and gene G_1 was calculated as follows:

$$HeteSim = \frac{1}{|O(C_1|CD)||I(G_1|DG)|} \sum_{i=1}^{|O(C_1|CD)|} \sum_{j=1}^{|I(G_1|DG)|} HeteSim(O_i(C_1|CD), I_j(G_1|DG))$$

where $O(C_1|CD) = \{D_1, D_2\}, I(G_1|DG) = \{D_1, D_2\}$. So the *HeteSim* value of C_1 and G_1 is 0.5.

In order to facilitate the calculation, the formula in Definition 3 can be normalized as Eq. (2):

Definition 4. NormHeteSim [10]

$$NormHeteSim\left(x_i, x_j|P\right) = \frac{M_{P_L}(x_i,:)M_{P_R^{-1}}(x_j,:)}{\sqrt{|M_{P_L}(x_i,:)||M_{P_R^{-1}}(x_j,:)|}} \tag{2}$$

where M is the relation matrix defined as follows:

Definition 5. Relation Matrix

$$M = U_{A_1A_2}U_{A_2A_3}\ldots U_{A_{l-1}A_l} \tag{3}$$

where $U_{A_iA_j}$ is the adjacency matrix A_i and A_j.

In addition, $M_P(i, j)$ represents the number of path instances of meta-path $P = (A_1A_2\ldots A_l)$ which start from object $x_i \in A_1$ to object $x_j \in A_l$, and $M_{P_L}(x_i,:)$ represents the feature vector of object x_i whose length is decided by the target object type A_l and $M_{P_R^{-1}}(b,:)$ represents the feature vector of object x_j, so the HeteSim value is the cosine similarity of the two feature vector. The path P_L and P_R in NormHeteSim definition is the decomposition of the path P from the middle position. That is, $P_L = (A_1A_2\ldots A_{mid})$ and $P_R = (A_{\text{mid}}A_{\text{mid}+1}\ldots A_l)$.

In biological heterogeneous network, measuring the similarity of drug and target is suited to the relevance search problem definition. Therefore, HeteSimQuery is considered baseline algorithm to measure the similarity between drug and target as Algorithm 1 shows. For any given meta-path which starts with drug type and end with target type, we can use HeteSimQuery to calculate the similarity of the query pairs (p, q). For example, if the similarity of one drug and gene pair under the path CGCDG (Compound-Gene-Compound-Disease-Gene) was queried, HeteSimQuery will return the similarity value of the query pair.

```
Algorithm 1. $HeteSimQuery(p,q)$
Input: data set $D$, query object pair $(p, q)$, meta-path $P$
Output: a similarity value of the query pair
Separate $P$ to $P_L$ and $P_R$
Calculate the Relation Matrix $M_{P_L}$ and $M_{P_R^{-1}}$
Generate Hash Vector of $p$ and $p$
Calculate the similarity of $(p, q)$ as Formula (2)
```

Obviously, when the inputing meta-path $P = (A_1 A_2 \ldots A_l)$ of HeteSimQuery is symmetrical or the starting object type is the same as the end object type, that is A_1 and A_l are the same object type, such as CDC or CDGDC, HeteSimQuery can be used to measure the similarity between objects with the same types. Therefore, our baseline algorithm HeteSimQuery can be used to measure the similarity between objects of the same types as well as different types.

3.3 Locality Sensitive Hash

Locality Sensitive Hash (LSH) functions were introduced to solve the approximate nearest neighbor problem in high dimensional spaces [14]. It is designed in such a way that if two objects are close in the intended distance measure, the probability that they are hashed to the same value is high, and if they are far in the intended distance measure, the probability that they are hashed to the same value is low [15]. Here the meaning of 'close' and 'far' depend on the similarity measure used, and the exact formulation of LSH functions varies with the exact distance definition of the similarity measure. Nevertheless, all LSH functions should always comply with the locality sensitive hashing schema [16].

However, not every similarity measure has its corresponding LSH functions satisfying locality sensitive hashing schema. Moses proposed a triangle inequality that existing LSH families satisfy [17]. Since HeteSim could eventually be placed under the cosine similarity of hash vectors, we just need to prove $\theta(x_i, x_j) \le \theta(x_i, y) + \theta(x_j, y)$: if x_i, x_j and y all lie in a plane, then it is obvious that the angle between x_i and x_j must be no greater than the sum of the angles between x_i and y and x_j and y; if x_i, x_j and y do not lie in a plane, and y' is the projection of y in the plane defined by x_i and x_j, then it holds that $\theta(x_i, x_j) \le \theta(x_i, y') + \theta(x_j, y')$. Note the sum of angles between x_i and y and x_j and y are greater than those between x_i and y' and x_j and y', so we have $\theta(x_i, x_j) \le \theta(x_i, y) + \theta(x_j, y)$. Therefore, HeteSim satisfies the locality sensitive hashing schema.

As the dataset scale of biological networks is always huge, and similarity computing is time-consuming, we use a relatively simple random hyperplane hash function as Definition 6 shows.

Definition 6. Hash Function

$$h_r(x) = \begin{cases} 1, & r \cdot x \geq 0 \\ 0, & r \cdot x < 0 \end{cases} \tag{4}$$

r is a d dimensional random vector with each value drawn from the standard Gaussian distribution $N(0,\ 1)$, if $r \cdot x \geq 0$, return 1, otherwise return 0. Then each d dimensional vector is hashed to one binary bit [16].

Given an integer m, we select m hash functions randomly and independently from the family defined in Definition 6, denoted as $H_m = \{h_{r_1}, \ldots, h_{r_m}\}$. By applying each of them to a dimensional vector x, we can map x to an m dimension vector in $\{0,\ 1\}^m$, denoted as $H_m(x)$. Then $H_m(x)$ is referred to the hash vector for x and m is the corresponding hash vector dimension. For any data set $D \in R^d, H_m(x)$ can generate a set consists of hash vectors, which is called the hash table of D. Given an integer t, we choose $H_m^1, H_m^2, \ldots H_m^t$ from H_m independently and randomly. Each hash family generates a $r_i (1 \leq i \leq t)$ dimensional hash table.

3.4 LSH-HeteSim

The biological networks dataset used in the prediction of interactions between drugs and targets is large-scale, usually consisted of numbers of biological databases. Moreover, mining the interactions between drugs and targets need large amount of similarity measure computations of high-dimensional vectors. These facts lead to that using the naive algorithms can be very time-consuming. Based on the characters above, we proposed an optimized algorithm based on LSH called LSH-HeteSim which can reduce a lot of similarity measure computation with the strategy that using a candidate subset generated by hashing reduces the computation times. As the relationship between the drug and the target is heterogeneous, the similarity measure we used in our method is HeteSim which is suitable for relevance search problem in heterogeneous networks as introduced in Sect. 3.2.

Algorithm 2. ***LSH-HeteSim***

```
Input: data set $D$, $m$, $t$, mea-path $P$, query object $p$
Output: a set of object pairs with their similarity value
Build LSH indexing for $D$ as 3.3 introduced
  for each hash table $T_{H_m^i} (1 \leq i \leq t)$ do
      Hashing query object $p$ to a bucket $B_i$
      Add objects hashed to the target bucket $B_i$ to set $Q$
  end for
  for each object $q \in Q$ do
      HeteSimQuery$(p, q)$
      Add all $(p, q)$ pairs and their similarity value into set $R$
  end for
Return $R$
```

The input data of Algorithm 2 is network dataset *D*, query object *p*, dimension of hash vector *m*, number of hash tables *t* and meta-path *P*, and the output result is a collection of objects which are in the candidate set with a similarity value. Firstly, Algorithm 2 creates a LSH indexing structure for given data set *D* (step 1); then it produces *t* hash tables cyclically, and eventually gets a collection of all objects which are mapped to the same hash bucket, as the candidate set *Q* (step 2–5); finally, it calculates the similarity between the query object *p* and each target object in candidate set *Q* using HeteSim, and returns a result set of all objects with a similarity value.

4 Experiments and Results

In order to completely inspect performance of our algorithm, we use several real biological networks as experiment data sets, rather than artificial data sets. Conducting experiments in real biological networks, it can direct compare capability and running time of our algorithm and existing methods.

4.1 Datasets

An integrated biological networks dataset Slap [6] is utilized as experimental material in this study. The network is constructed from 17 public data sources, which contains 305,792 nodes and 670,546 edges (Fig. 1). For the network, nodes are categorized into 10 types, in which 11 connections are existed. In addition, a single node is an instance of a corresponding type, for example: a node for drug Clofarabine (CID: 119182, Molecular Formula: C10H11ClFN5O3) is an instance of type Chemical Compound. A path is an instance of a corresponding meta-path, defined in Definition 2, for example: Troglitazone-Disease(776)-VEGE is an instance of the meta-path: Compound-Disease-Genes, here Troglitazone is a drug that can treat the disease 776 caused by VEGE.

4.2 Effectiveness

Assessing Drug Similarity. In Sect. 3.2, we have already mentioned when we choose the right meta-path which is to ensure the starting object type is the same as end object type, such as drugs, HeteSimQuery can be used to measure the similarity between objects with the same type. Here we use HeteSimQuery to cluster drugs. We took 40 kinds of drugs from 4 disease areas (headache, diabetes, HIV and asthma) to determine whether our method is able to distinguish drugs from different therapeutic areas. For each drug, we calculated its similarity value with the other 39 drugs. Then we selected all the drug-drug pairs whose similarity values were greater than a predefined threshold. In practice, drug-target interactions were visualized by the Cytoscape software [19], and functions of drug target genes were annotated through the iGepros server [21].

Since the path we used is CDGDC, namely two drugs can be regarded as similar when they can treat diseases caused by same genes. Therefore drugs related to same kind

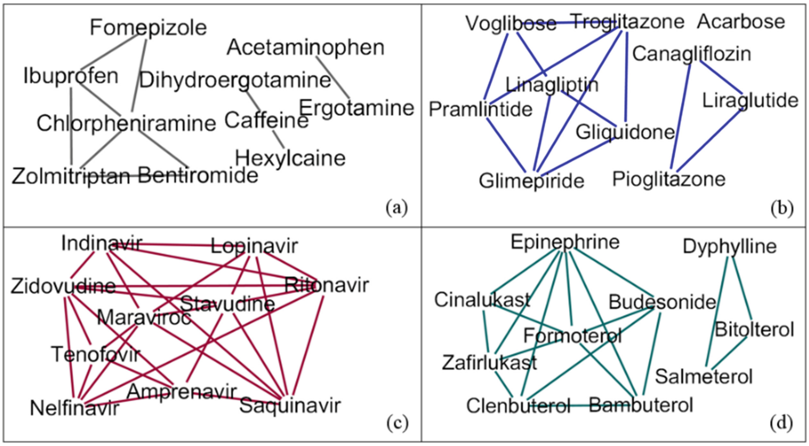


Fig. 3. Drug similarity network

of diseases have a higher probability to connect together. Our experiment results support this viewpoint. As shown in Fig. 3, the drugs in each group tend to have an ability to treat the same kind of disease, for example, the grey-colored drugs like Ibuprofen, Chlorpheniramine, Fomepizole can treat the headache and they connected together.

Comparison with SLAP. To further evaluate the drug-target interaction mining effect of the our optimized algorithm LSH-HeteSim, we compare LSH-HeteSim with the baseline algorithms HeteSimQuery and SLAP respectively. The experimental dataset used here is a subset extracted from the Slap dataset, including 1000 drugs, 127 targets and 3762 drug-target links. Then we have 127,000 drug target pair samples (3762 positive and 123,238 negative samples). The algorithm HeteSimQuery and SALP need to compute the similarities over the whole 127,000 drug-target pairs. However, LSH-HeteSim only needs to compute similarity for each query object with its corresponding candidate set. In our experiments, the parameter *m* and *t* are assigned 20 and 5 respectively, therefore LSH-HeteSim takes 22,170 times similarity computation in total.

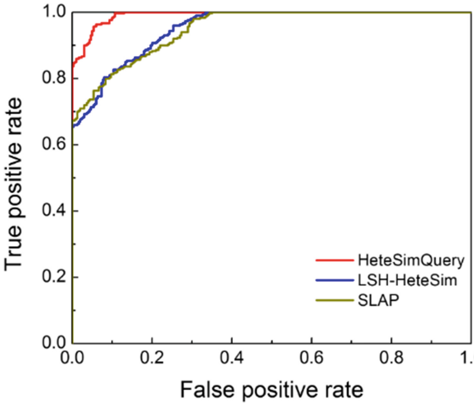


Fig. 4. ROC curves among different methods

To compare the accuracy effects of these three algorithms, we performed a ROC [20] (receiver operating characteristics) statistical analysis over the results. The ROC curves are shown in Fig. 4 which present achievable true positive rates (TP) with respect to all false positive rates (FP). The *AUC* (area under an ROC curve) values of HeteSimQuery, SLAP and LSH-HeteSim are 0.982, 0.940 and 0.943 respectively. Obviously, these three algorithms all have good prediction accuracy in the dataset. However, compared with our LSH-HeteSim algorithm, HeteSimQuery and SLAP algorithms require more running times. We will compare and describe in the experiment of Sect. 4.3. For drug target query in the large-scale biological data, time efficiency is very important. Our LSH-HeteSim algorithm reduced the similarity calculation times of high dimensional vector by LSH so that it can reduce the query time while ensure the prediction accuracy.

4.3 Efficiency

Efficiency is measured by the running time of algorithm. Since the hash functions are randomly picked, each experiment is repeated 10 times and the average is reported. The input of LSH-HeteSim algorithm has two parameters: the hash vector dimension *m*, the number of hash tables *t*. Here we take *m* and *t* into account to discuss how the running time changes.

As HeteSim is a path-dependent method, the running time is various when different path is selected. In out experiment, we use the meta-path: CGCDG (Compound-Gene-Compound-Disease-Gene). To better describe the running time with different parameters, we divide the experiments into two groups, and discuss the impact of parameters *t* and *m* on the running time.

Firstly, we randomly select a compound (noted as CID) as the query object such as CID = 5880, and the parameter *m* was assigned 20, then run programs for *t* = 1, 2, 3, 4, 5 respectively. Results of experiment are shown in Fig. 5(a). Clearly the trend of curve can be seen from the diagram, running time increases with the value of the parameter *t* added, mainly because a bigger value of t means more hash tables, and therefore more time to require for calculation.

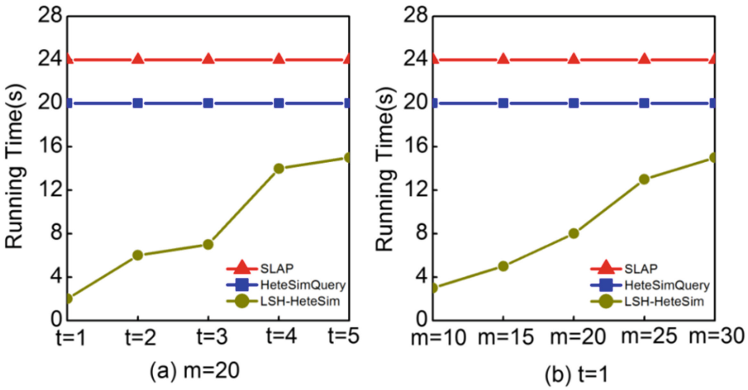


Fig. 5. Running time on Slap

Then we set parameter t as 1 and let the value of parameter m be 10, 15, 20, 25, and 30 respectively. Each experiment was repeated 10 times to take the average running time as the result. Obviously with the parameter m increases, the running time is growing, and this is mainly due to the increase in dimension of the random vector, it will take more time to calculate the similarity of two objects.

For a certain query object, the running times for SLAP and HeteSimQuery are almost fixed and more than LSH-HeteSim's, so their cures are linears. It is mainly due to our candidate sets generated by LSH, which greatly reduce the computation times of high-dimensional vectors. Considering results of effectiveness and efficiency assesses, the LSH-HeteSim algorithm has less running time compared with the state-of-art methods, and its prediction accuracy is still comparable to these methods. In addition, there is no fixed range of the two parameters, and the suitable values of the two parameters should be assigned through value trials. With the increase of the parameters m and t, the times of similarity calculations and the dimension of feature vectors increase. Despite the prediction accuracy will rise, however, the running time is also rapidly increasing.

5 Discussion and Conclusion

For the LSH-HeteSim algorithm, it accelerates computing of similarity search in high-dimensional space through the LSH method, which results in a slightly decrease of search accuracy (experiments in Fig. 4). In the future, we will use the MP-LSH method [22] instead of the LSH method to optimize our algorithm. In this way, accuracy of the LSH-HeteSim algorithm can be improved, and its less running time characteristics can be kept. In addition, the meta-path CGCDG used in this study is based on certain biological information, while meta-paths coming from other biological information should be inspected in the future.

In this study, we proposed an efficient drug-target interaction mining algorithm for heterogeneous biological networks called LSH-HeteSim. Experiment results show that our proposed algorithm can effectively predict interactions between drugs and targets. Specially, for larger-scale biological data, LSH-HeteSim has less running time compared with the state-of art methods.

Acknowledgment. The work was supported in part by the National Natural Science Foundation Project of China under Grant. No. 61170096 and Research Program of Shanghai Science and Technology Committee under Grant. No. 12511502403. The authors gratefully acknowledge the support of SA-SIBS Scholarship Program.

References

1. Hanzlik, R.P., Koen, Y.M., Theertham, B., et al.: The reactive metabolite target protein database (TPDB)—a web-accessible resource. BMC Bioinf. **8**(1), 95 (2007)
2. Chan, S.Y., Loscalzo, J.: The emerging paradigm of network medicine in the study of human disease. Circul. Res. **111**(3), 359–374 (2012)

3. Chen, L., Lu, J., Luo, X., et al.: Prediction of drug target groups based on chemical-chemical similarities and chemical-chemical/protein connections. Biochim. Biophys. Acta (BBA)-Proteins and Proteomics **1844**, 207–213 (2013)
4. Campillos, M., Kuhn, M., Claude, G., et al.: Drug target identification using side-effect similarity. Science **321**(5886), 263–266 (2008)
5. He, Z., Zhang, J., Shi, X.H., et al.: Predicting drug-target interaction networks based on functional groups and biological features. PloS one **5**(3), e9603 (2010)
6. Chen, B., Ying, D., David, J.W.: Assessing drug target association using semantic linked data. PLoS Comput. Biol. **8**(7), e1002574 (2012)
7. Yamanishi, Y., Kotera, M., Kanehisa, M., et al.: Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework. Bioinformatics **26**(12), 246–254 (2010)
8. Fakhraei, S., Louiqa, L., Lise, G.: Drug-target interaction prediction for drug repurposing with probabilistic similarity logic. In: Proceedings of the 12th International Workshop on Data Mining in Bioinformatics. ACM (2013)
9. Sun, Y.Z., Han, J.W., Yan, X.F., et al.: PathSim: meta path-based top-k similarity search in heterogeneous information networks. In: VLDB'11 (2011)
10. Shi, C., Kong, X.N., Yu, P.S., et al.: Relevance search in heterogeneous networks. In: Proceedings of the 15th International Conference on Extending Database Technology. ACM (2012)
11. Palma, G., Viadl, M.-E., Haag, L., et al.: Measuring relatedness between scientific entities in annotation datasets. In: Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics. ACM (2013)
12. Getoor, L., Diehl, C.P.: Link mining: a survey. ACM SIGKDD Explor. Newslett. **7**(2), 3–12 (2005)
13. Yu, H.Y., Braun, P., Yildirim, M.A., et al.: High-quality binary protein interaction map of the yeast interactome network. Science **322**(5898), 104–110 (2008)
14. Datar, M., Immorlica, N., Indyk, P., et al.: Locality-sensitive hashing scheme based on p-stable distributions. In: Proceedings of the Twentieth Annual Symposium on Computational Geometry. ACM (2004)
15. Jegou, H., Matthijs, D., Cordelia, S.: Product quantization for nearest neighbor search. IEEE Trans. Pattern Anal. Mach. Intell. **33**(1), 117–128 (2011)
16. Kishore, S.: Accelerated clustering through locality-sensitive hashing. Diss. Massachusetts Institute of Technology (2012)
17. Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: Proceedings of the thirty-fourth annual ACM symposium on Theory of computing. ACM (2002)
18. SLAP for Drug Target Prediction. http://cheminfov.informatics.indiana.edu:8080/slap
19. Saito, R., Smoot, M.E., Ono, K., Ruscheinski, J., Wang, P.L., Lotia, S., Pico, A.R., Bader, G.D., Ideker, T.: A travel guide to Cytoscape plugins. Nat. Methods **9**(11), 1069–1076 (2012)
20. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. **27**(8), 861–874 (2005)
21. Zheng, G., Wang, H., Wei, C., Li, Y.: iGepros: an integrated gene and protein annotation sever for biological nature exploration. BMC Bioinf. **12**(Suppl 14), S6 (2011)
22. Lv, Q., Josephson, W., Wang, Z., et al.: Multi-probe LSH: efficient indexing for high-dimensional similarity search. VLDB Endowment, pp. 950–961 (2007)

Comparison of Hemagglutinin and Neruaminidase of Influenza A Virus Subtype H1N1, H5N1, H5N2, and H7N9 Using Apriori Algorithm

Dae Young Kim(✉), Hye-Jun Kim, Junhyeok Bae, and Taeseon Yoon

Hankuk Academy of Foreign Studies, Mohyeon-myeon,
Cheoin-gu, Yongin-si, Gyeonggi-do, South Korea
{dae8177, lilliput222, bjh5098, tsyoon}@naver.com

Abstract. The Spanish flu first occurred in 1918 and killed about 50 million people in the world. In 2005, by using gene decoding process, Robert B. Belshe identified that the Spanish flu was occurred by H1N1, which is highly pathogenic influenza A virus. Influenza A virus has been mutated consistently and unexpectedly; H5N1, H5N2, and H7N9 which used to be known as not human infecting virus now infect a lot of people in the world. In this research, by using Apriori Algorithm, we compared amino acid strain of Hemagglutinin and Neuraminidase, which are glycoprotein of H1N1, H5N1, H5N2, and H7N9 to figure out their similarity in amino acid strain. Furthermore, in the case of H7N9, we also compared the amino acid data from 1988, 2009 (bird), and 2013 (human) and proposed the significance of biological research about the site which differs in terms of amino acids.

Keywords: Spanish flu · Influenza A Virus Subtype H1N1, H5N1, H5N2, H7N9 · Hemagglutinin · Neuraminidase · Apriori algorithm

1 Introduction

In March 1918, as Albert Gitchell, a company cook, got infected with influenza virus, the number of new infections increased at an alarming rate. Within a year, 500 million people were infected and at least 50 million people died. They are all victims of the Spanish Flu.

The autopsy result revealed that the victims had Cytokine Storm in common which had caused excessive immune response that damaged pulmo and therefore had shown Acute Respiratory Syndrome [1].

According to the report made by Robert B. Belshe in 2005, we can infer that the virus which caused the Spanish flu is H1N1. References [2, 3] Highly pathogenic Influenza A subtype H1N1 has continuously mutated and caused the 1957 Asian Flu (H2N2) epidemic and the 1968 Hong Kong Flu (H3N2) epidemic. Also, the outbreak of Avian Influenza H5N1 which crossed the barrier between species (humans-birds) was reported in 15 countries.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 77–86, 2014.
DOI: 10.1007/978-3-319-13186-3_8

It is important to note that although H5N1 is not contagious thorough air compared to H1N1, the symptom that follows, Cytokine Storm, is identical to Spanish Flu [4]. As shown above, H5N1 which was previously predicted to be non-contagious regarding humans is showing a tendency to make humans as hosts over the course of time. Also, H5N2 which is less contagious than H1N1 was reported to have infected a human in Japan on January 2006. Furthermore, on March 2013, 3 people who live Shanghai and Anhui were infected by the virus H7N9 and 2 of them died. Actually, the infection of H7N9 among the human species has never occurred before [5, 6]. Only two weeks after the outbreak (April 13, 2013), 49 people were infected and 11 of them died.

Therefore, in order to compare the similarities among the amino acid structure of the respective Hemagglutinin and Neuraminidase of H1N1, H5N1, H5N2, and H7N9 we used Apriori Algorithm. The FASTA form data of the amino acid structure used in the analysis was from NCBI (National Center for Biotechnology Information). In addition, to propose a research subject about the mutation tendency of H7N9, which was occurred recently, we also compared the H7N9 fowl samples of 1988, 2009, and human's of 2013 using the Apriori Algorithm.

2 Influenza Virus

2.1 Type A, B, and C

Influenza can be classified into 3 types; type A, B, and C [9]. Type A Influenza virus is infectious to some mammals, including fowl, human and swine, and 19 kinds of them were discovered at present. Since it has variety of hosts, their possibility of global pandemic is relatively high. Furthermore, since some mammals, such as swine, has more than one virus acceptors on their respiratory epithelial cells, the virus can be carried through these mammals which acts as a medium to human and be combined with human's acceptor. In other words, type A virus can surpass the barrier between species. On the other hand, only one kind of virus exists in Type B and Type C so they are not highly transmissible. In addition, it is currently known that Type B virus only infects human and seal, and Type C virus only infects human and swine. However, all 3 types of viruses' overall structure are same.

2.2 Hemagglutinin and Neuraminidase

Influenza A viruses has major 11 genes on 8 RNA fragment. These genes encode 11 protein information (Hemagglutinin, Neuraminidase, NP, M1, M2, NS1, NS2 (NEP), PA, PB1, PB1-F2, and PB2) and the dividing criteria for Influenza A virus is the type of Hemagglutinin and Neuraminidase. For that reason, Influenza A virus subtypes are named using H and N; H is numbered from 1 up to 16 and N is numbered from 1 up to 9 (ex. H1N1, H5N1 …). Hemagglutinin and Neuraminidase are both glycoprotein which play a pivotal role in virus infection and replication. Hemagglutinin initiates virus infection by combining with sialic acid from epithelial cell. Neuraminidase helps separating the combination between the virus and the host cell made by Hemagglutinin, so the replicated viruses can be released out of the cell.

2.3 Cytokine Storm

Influenza A virus subtype H1N1 and H5N1 gave rise to cytokine storm to patients in common. Cytokine is a generic name of proteins which act as hormonal signaling molecules that stimulates the immune system. Cytokine Storm is a fatal immune reaction caused by highly elevated levels of Cytokine leading to MODS (Multiple Organ Dysfunction Syndrome) such as lung damage. In other words, positive feedback loop between cytokines and immune cells arouses excessive immune response which cause disorder in physical functions. As shown in Fig. 1, when a pathogen invades and infects macrophages, the macrophage delivers its information that triggers hyper secretion of cytokine by the immune reaction and pulmonary disorder occurs consequently [11].

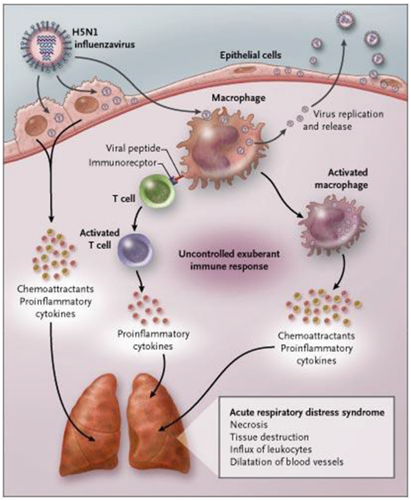


Fig. 1. The mechanism of cytokine storm evoked by influenza virus (Source: Osterholm. New England Journal of Medicine, 352 (18): 1839, Fig. 3. May 5, 2005.)

Because of the Cytokine Storm, the casualties of the youth with better immunity were bigger than the aged with poor immunity. Avian flu (H5N1), 1957 Asian Flu (H2N2), 1968 Hong Kong Flu (H3N2), 1918 Spanish Flu and 2009 Mexican Swine Flu (H1N1) were caused by those different viruses but have one thing in common; all of them provoked cytokine storm to victims with a certain degree.

3 Correlation Between Protein Structure and Its Functions

3.1 Protein Three-Dimensional Structure

Glycoprotein on the surface of Influenza A Virus is protein after all. Therefore, it has its own peculiar three-dimensional structure which is also called as tertiary protein structure. Tertiary protein structure is built as irregular structures in α-helix, β-sheet, and coil structure, the secondary protein structure, are folded. Studies have shown that Tertiary protein structure determines its functional properties of proteins. Thus comprehending three dimensional structure of glycoprotein in Influenza A virus whose attributes are under control of the types of surface glycoprotein is significant [14–16].

3.2 Secondary Structure and Three-Dimensional Structure

There are various methods in predicting 3D structure of protein and the most popular method among them is template based modeling. This modeling is done by finding certain proteins from the template whose functions are similar to the target protein (the protein we want to know) using amino acid sequence. The key point of this method is that the proteins with similar amino acid sequence have similar 3D structure. In fact, protein structure is generally called similar if the index of similarity of amino acid sequence is over 20–30 %. To model a protein whose index of similarity of amino acid is lower than 20 %, ab initio modeling has to be done. It uses energy function about protein structure but its precision is relatively lower than template based modeling.

4 Experiment

4.1 Data

FASTA form of Hemagglutinin and Neuraminidase of H1N1, H5N1, H5N2, and H7N9 was used to analyze on the basis of the number of amino acid frequency. Those FASTA data of H5N1 and H5N2 viruses were found in fowls and the data of H1N1 virus was found in humans.

Furthermore, to investigate the mutation trend of H7N9 with time, amino acid data of Hemagglutinin and Neuraminidase from turkey in 1988, goose in 2009 and Chinese in 2013 were used. All of these experimental data was obtained from National Center for Biotechnology Information (NCBI) [7].

4.2 Algorithm

The algorithm used in the comparison of the amino acids is Apriori. This Algorithm is more efficient than Artificial Neural Network or Support Vector Machine in that Apriori can estimate the correlation among data frequency. Apriori is an association rule discovery algorithm used to systematically control the geometrical growth of candidate item set by using support-based pruning. The basic principle of Apriori starts from the supposition that if one set of the data frequently occurs, its subset also frequently occurs. The process of creating frequent-data-set, creating subset, pruning subset, calculating support level is repeated in this order (see Fig. 2).

Since Apriori algorithm searches for frequency of the dataset and calculate its support level, it allows us to assess the dispersion of the given dataset easily and

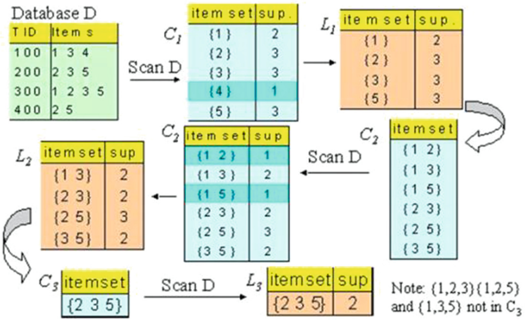


Fig. 2. The process of Apriori Algorithm

quickly. Thus, we decided that the Apriori algorithm is the optimized algorithm for analyzing amino acid sequences and used it for this research.

Window size is a point to be considered so the experiments were divided into environments of window 5, 7 and 9. Particularly, to compare H7N9 on a year-on-year basis, we used WEKA program.

5 Results

5.1 H1N1, H5N1, and H5N2

For better interpretation, we neglected data whose accuracy (≤1) was smaller than 0.8. Figures 3, 4, and 5 is the result of the experiments done in Window 5, 7, and 9, respectively. The x-axis of each graph is the type of amino acid and the y-axis is the appearance frequency for each amino acids. The more appearance frequencies among the viruses are alike, the better we can estimate that their structure resembles each other.

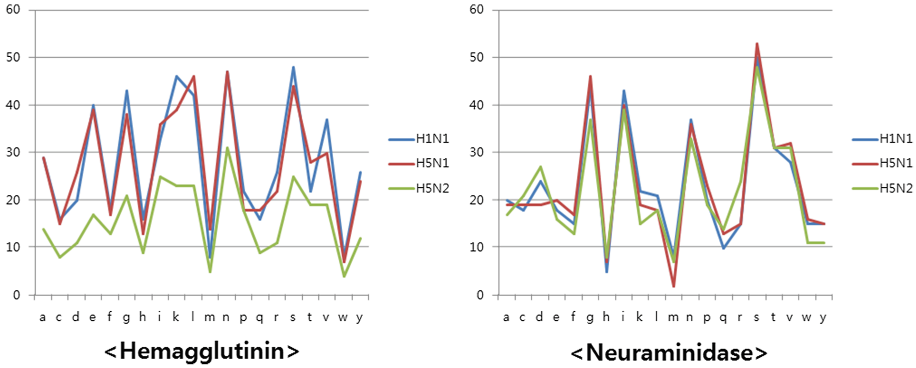


Fig. 3. The graph of H1N1, H5N1, and H5N2 amino acid frequency in window size 5

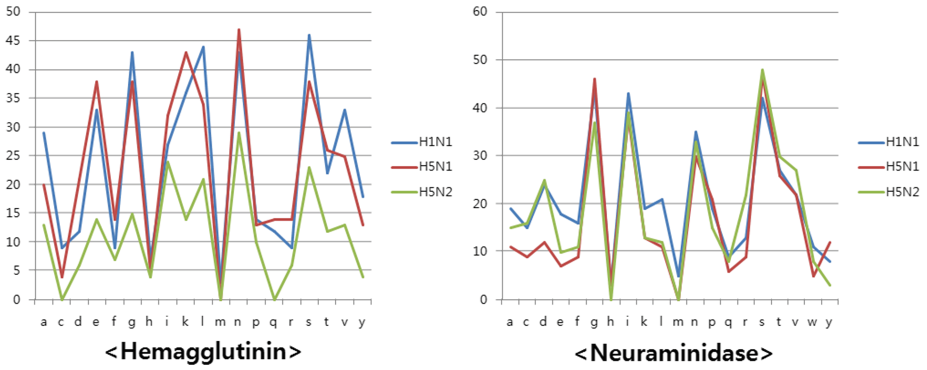


Fig. 4. The graph of H1N1, H5N1, and H5N2 amino acid frequency in window size 7

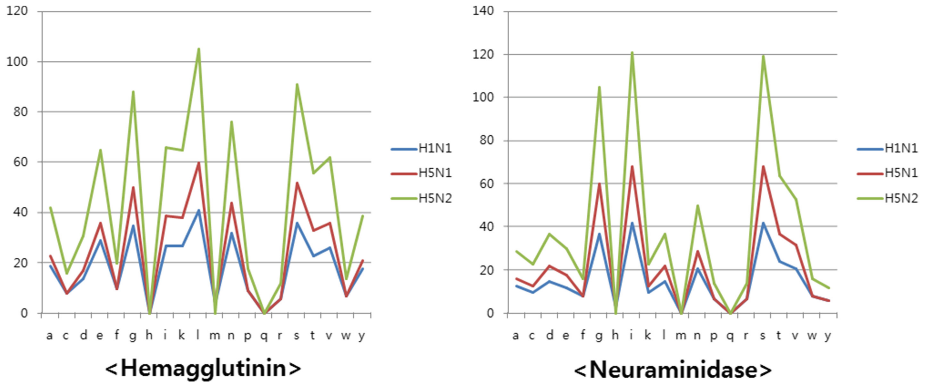


Fig. 5. The graph of H1N1, H5N1, and H5N2 amino acid frequency in window size 9

Particularly, in the experiment done in window 5 and 7 (Figs. 3 and 4), the similarity among the appearance frequency of Hemagglutinin and Neuraminidase in H1N1, H5N1 and H5N2 was exceedingly high.

5.2 H7N9

Similar to previous experiment, we neglected data whose accuracy (≤1) was smaller than 0.8. Figures 6, 7, and 8 are the results of the experiments done in Window 5, 7, and 9, respectively.

Figures 6, 7, and 8 show that the amino acid strain of H7N9 virus's glycoprotein from 2009 goose and 2013 Chinese are very similar even if they are from different host species. The overall distribution of amino acid data is quite similar regardless of years and window size but in the perspective of the specific frequency numbers of each amino acid, the data from 1988 turkey are relatively different.

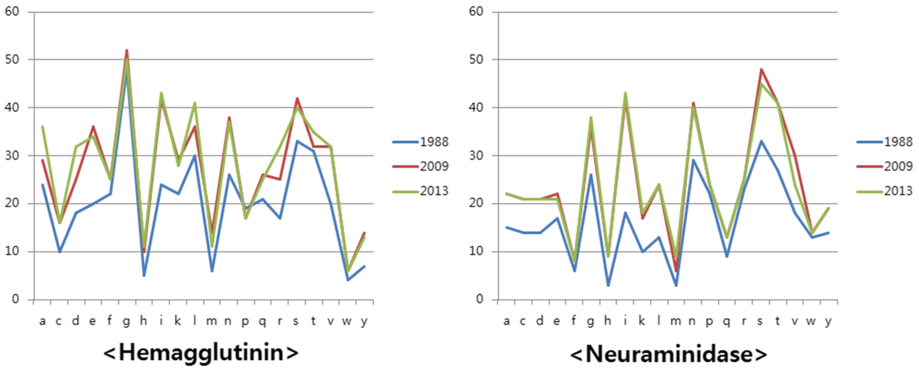


Fig. 6. The graph of H7N9 amino acid frequency in window size = 5

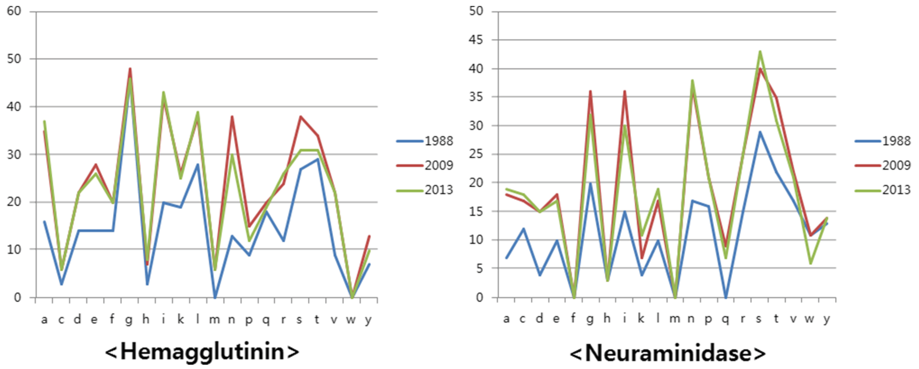


Fig. 7. The graph of H7N9 amino acid frequency in window size = 7

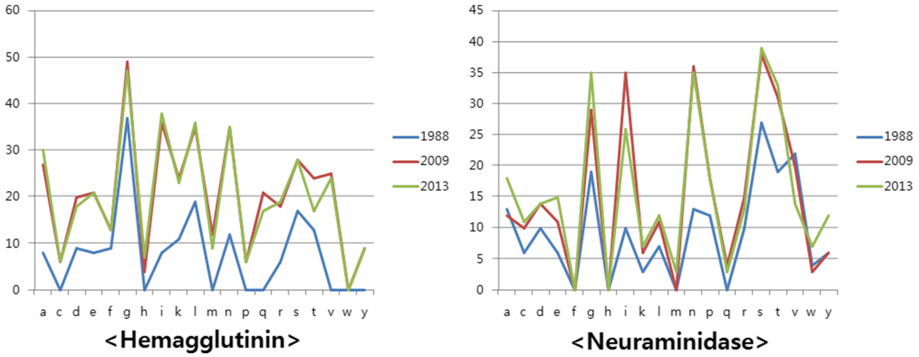


Fig. 8. The graph of H7N9 amino acid frequency in window size = 9

5.3 H1N1, H5N1, H5N2, and H7N9

We conducted the experiment in same condition with previous experiment. Figures 9, 10, and 11 are the results of experiments done in Window 5, 7, and 9, respectively.

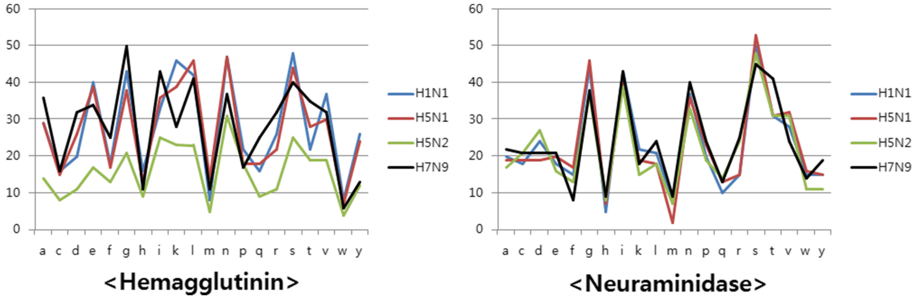


Fig. 9. The graph of H1N1, H5N1, H5N2, and H7N9 amino acid frequency in window size 5

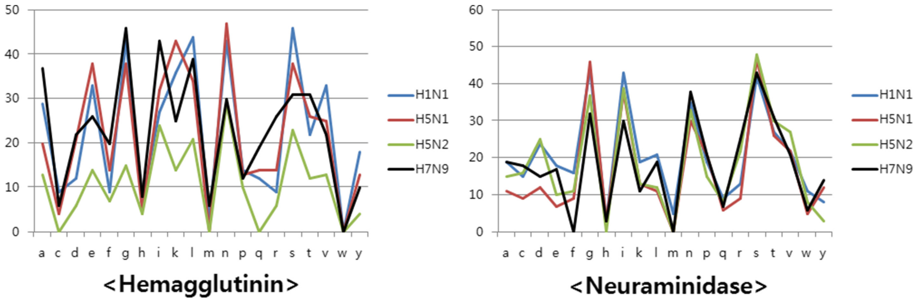


Fig. 10. The graph of H1N1, H5N1, H5N2, and H7N9 amino acid frequency in window size 7

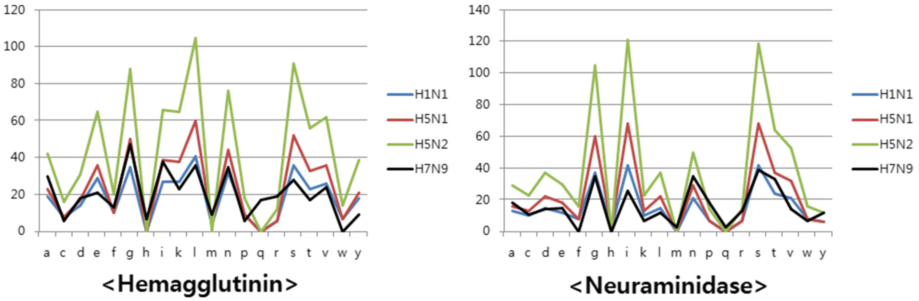


Fig. 11. The graph of H1N1, H5N1, H5N2, and H7N9 amino acid frequency in window size 9

6 Discussion

6.1 The Similarity of H1N1, H5N1, and H5N2

From the results of ***4.1 H1N1, H5N1, and H5N2***, the similarity between those three viruses' glycoprotein amino acid strain is measured high. The similar structure of amino acid means that the proteins will do similar biological function. Especially, in the case of Influenza virus, the proteins located outer surface of lipid envelope such as Hemagglutinin or Neuraminidase conduct crucial function of viruses' infection and replication process. H1N1 in 1918 Spanish Flu was spread extremely fast and caused deadly disease with cytokine storm. H5N1 and H5N2 are avian flu and although the latter is not highly pathogenic, H5N1 is could be dangerous to humans; it is relatively easy to mutate into "H1N1-like" virus since their features of causing cytokine storm to victims are same and their amino acid structure of glycoprotein are similar. In other words, if it mutates and becomes ideal to spread fast through the air and being well transmitted between humans, the possibility of outbreak of catastrophe would increase. Therefore, yet there is no suitable ways to delay or stop the Influenza viruses' mutation, we should struggle to find the prevention of disaster and to predict the viruses' mutation more exactly.

6.2 The Mutation of H7N9

From the result of ***4.2 H7N9***, we can infer that even if it shows high similarities as time pasts, Influenza A virus subtype H7N9 has been mutated to "a certain slowly" since 1988. Finally, it has changed its features nowadays and became deadly to humans. Also, there are some amino acids which do not have similar number of frequency through three kinds of years, especially in 1988. For example, in Fig. 7, which is the result of experiment with window size 9, the amino acid "V" shows relatively big difference between 1988 data and the others, either in Hemagglutinin or Neuraminidase. This suggests the structure that amino acid V (Valine) is contained has changed which affected the feature of 1988 H7N9.

6.3 Overall Comparison

The result of ***4.1 H1N1, H5N1, and H5N2*** shows some similarities between viruses that cause serious cytokine storm (H1N1, and H5N1) or those which are avian influenza (H5N1, and H5N2). The result of ***4.2 H7N9*** suggests some mutational tendency of H7N9. To consider the result in wide perspective, we compared all four viruses (***4.3 H1N1, H5N1, H5N2, and H7N9***). They are in common that they are Influenza A virus subtypes which infect human beings and threat our lives. As Figs. 8, 9, and 10 show, four viruses' glycoprotein amino acid structure are similar in abstract shape. By interpreting those results by more elaborate experiment, we can infer some mutational tendency of Influenza A viruses more exactly.

References

1. Watanabe, T., Watanabe, S., Shinya, K., Kim, J.H., Hatta, M., Kawaoka, Y.: Viral RNA polymerase complex promotes optimal growth of 1918 virus in the lower respiratory tract of ferrets, PNAS 2009
2. Belshe, R.B.: The origins of pandemic influenza-lessons from the 1918 virus. NEJM **353**, 2209–2211 (2005)
3. Taubenberger, J.K., Morens, D.M.: 1918 Influenza: the mother of all pandemics. Emerg. Infect. Dis. **12**, 15–22 (2006)
4. U.S. Department of Health & Human Services. http://www.flu.gov/about_the_flu/h5n1/
5. Notification of three human cases of H7N9 in Shanghai and Anhui. http://www.info.gov.hk/gia/general/201303/31/P201303310295.htm
6. Article Dozens In Japan May Have Mild Bird Flu, January 2006. http://www.cbsnews.com/news/dozens-in-japan-may-have-mild-bird-flu/
7. National Center for Biotechnology Information. http://www.ncbi.nlm.nih.gov
8. Taubenberger, J.K.: The origin and virulence of the 1918 "Spanish" influenza virus. Proc. Am. Philos. Soc. **150**, 86–112 (2006)
9. USGS National Wildlife Health Center: The avian Influenza H5N1 Threat – Current facts and future concerns about Highly Pathogenic avian Influenza H5N1, p. 2, August 2005
10. Rabadan, R., Robins, H.: Evolution of the influenza a virus: some new advances. Evol. Bioinform. Online **3**, 299–307 (2007)
11. Osterholm, M.T.: Preparing for the Next Pandemic. N. Engl. J. Med. **352**, 1839–1842 (2005)
12. Johnson, A.P.: RN, MPH, CHES (Northwest Ohio Consortium for Public Health): CYTOKINE STORM and the INFLUENZA PANDEMIC (2005)
13. Simonsen, L., Clarke, M.J., Schonberger, L.B., Arden, N.H., Cox, N.J., Fukuda, K.: Pandemic versus epidemic influenza mortality: a pattern of changing age distribution. J. Infect. Dis. **178**, 53–60 (1998)
14. Hegyi, H., Gerstein, M.: The relationship between protein structure and function: a comprehensive survey with application to the yeast genome. J. Mol. Biol. **288**, 147–164 (1999). Department of Molecular Biophysics and Biochemistry, Yale University, New Haven, CT, USA
15. Warshel, A.: Electrostatic basis of structure – function correlation in proteins. Acc. Chem. Res. **14**, 284–290 (1981). Department of Chemistry, University of Southern California, Los Angeles, CA
16. Hellinga, H.W.: Computational protein engineering. Nat. Struct. Biol. **5**, 525–527 (1998)
17. Berg, J.M., Tymoczko, J.L., Stryer, L.: The amino acid sequence of a protein determines its three-dimensional structure (Sect. 3.6). In: Berg, J.M., Tymoczko, J.L., Stryer, L. (eds.) Biochemistry, 5th edn. W H Freeman, New York (2002)
18. Hellinga, H.W.: Rational protein design: combining theory and experiment. Proc. Natl. Acad. Sci. U.S.A. **94**, 10015–10017 (1997)

Privacy Preserving Association Rule Mining Using Binary Encoded NSGA-II

Peng Cheng, Jeng-Shyang Pan(✉), and Chun-Wei Lin

Shenzhen Graduate School, Harbin Institute of Technology,
Shenzhen 518055, People's Republic of China
surf_mailbox@163.com, jengshyangpan@gmail.com,
jerrylin@ieee.org

Abstract. When people utilize data mining techniques to discover useful knowledge behind a large database; they also have the requirement to preserve some information so as not to be mined out, such as sensitive or private association rules, classification tree and the like. A feasible way to address this problem is to sanitize the database to conceal sensitive information. In this paper, we focus on privacy preserving in association rule mining. In light of the tradeoff within the side effects accompanying the hiding process, we tackle this problem from a point view of multi-objective optimization. A novel association rule hiding approach was proposed based on evolutionary multi-objective optimization (EMO) algorithm. The binary encoding scheme was adopted in the EMO algorithm. Three side effects, including sensitive rules not hidden, non-sensitive lost rules and spurious rules were formulated as objectives to be minimized. The NSGA II algorithm, a well established EMO algorithm, was utilized to find a suitable subset of transactions to modify by removing items so that the three side effects are minimized. Experiment results were reported to show the effectiveness of the proposed approach.

Keywords: Privacy preserving data mining · Association rule mining · Evolutionary multi-objective optimization · EMO

1 Introduction

Nowadays, data often need to be shared among different organizations during business collaboration. People can utilize data mining techniques to extract useful knowledge from a large data collection. However, this also brings the risk of disclosing sensitive knowledge to other parties. Verykios et al. in [1] introduced an example to illustrate the need to share data and hide sensitive rules. In order to balance the confidentiality of the disclosed data with the legitimate mining needs, the original database can be modified in some way so that the sensitive knowledge can be hidden when data is shared or published. However, such modification could lead to conceal non-sensitive knowledge or to generate new ghost rules. So the challenge is how to minimize the side effect along with the modification.

Here we specially focus on privacy preserving on association rules mining. Association rule mining is a commonly used technique in the data mining field. Association rules hiding refer to modification on original database in some ways so that

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 87–99, 2014.
DOI: 10.1007/978-3-319-13186-3_9

certain sensitive rules existent in the original database can't be mined out in the modified database. It belongs to the field of data sanitization. Since the database is changed, when applying the same association rule mining algorithm with the same parameters setting on the modified database, the obtained rules set could be different. After sanitization, some rules might be lost and some new rules might be added. Three criteria can be used to assess the side effects the hiding process brings. They are sensitive rules not be hidden, non-sensitive rules lost and new ghost rules generated after sanitization.

Some algorithms were proposed to solve the problem of association rules hiding [7]. Most are exact or deterministic. Atallah et al. [2] proved the optimal solution to this problem is NP-hard and first proposed a protection algorithm for it. Dasseni et al. [1, 3] extended the itemset hiding to association rules and proposed heuristic hiding approaches, i.e., algorithm *1.a*, *1.b*, *2.a*, *2.b* and *2.c*. These approaches hide sensitive rules by deleting or inserting items in a database to decrease the supports or confidences of sensitive rules below the specified thresholds. One of the assumptions of their approaches is that sensitive rules are disjoint. Amiri [6] proposed heuristic algorithms to hide itemset (not rules) by removing transactions or items, in terms of the number of sensitive and non-sensitive itemsets related. Wu et al. [4] designed a method aimed at avoiding all the side effects in rule hiding process instead of hiding all sensitive rules. Sun [8] proposed a border-based approach, which focused on preserving the border of non-sensitive frequent itemsets rather than considering all during the sanitization process. Based on this concept, Divanis et al. [9] proposed an exact approach which hides rules by extending the database. Borrowing the concept of TF-IDF (Term Frequency-Inverse Document Frequency) in the field of text mining, Hong et al. [10] devised a greedy-based hiding approach which assigns each transaction a SIF-IDF value to evaluate the correlation degree of the transaction with the sensitive itemset. The supporting transactions are sorted by SIF-IDF values and the candidates with highest values are selected to modify by removing items.

In this paper, we solved the rules hiding problem using binary encoding based evolutionary multi-objective optimization (EMO) algorithm and three side effects were formulated as optimization goals to be minimized. The model we adopted to modify database and hide rules is to delete some items in identified transactions in such a way that sensitive rules escape the mining in the modified database at some predefined thresholds, while the three side effects are minimized as much as possible. The EMO algorithm is used to find suitable transaction candidates to modify. Through a set of experiments, we demonstrated the effectiveness of this approach.

2 Problem Formulations

We first introduce the basic notions about association rules mining and hiding. Table 1 summarizes the notations used in this paper.

A rule $X \rightarrow Y$ is strong if it satisfies the following condition: (1) $Supp(X \cup Y) \geq MST$ and (2) $Conf(X \rightarrow Y) \geq MCT$. With given *MST* and *MCT*, the task of association rule mining is to find all frequent itemsets and strong rules from database *D*.

Table 1. Notations and definitions

Notation	Definition
I	$I = \{I_1, I_2, \ldots, I_m\}$ is a set of items available
Itemset	An itemset is a subset of I
t	The transaction t is denoted as $t = <ID, X>$, where ID is a unique transaction identifier number and X represents a list of items making up the transaction
rule	The association rule $X \rightarrow Y$ means that the antecedent X infers to the consequent Y. Here both X and Y are itemsets. $X \cap Y = \emptyset$
$Supp(X)$	The relative support of X. It is the fraction (or percentage) of the transactions in database which contain itemset X
$Conf (X \rightarrow Y)$	The confidence of the rule $X \rightarrow Y$ is computed as $Supp(X \cup Y)/Supp(X)$. It indicates a rule's reliability
MST	The minimum relative support threshold specified by user
MCT	The minimum confidence threshold specified by user
D	The original transactional database. $D = \{t_1, t_2,.., t_n\}$
D'	The sanitized/modified/released database, which is transformed from D
R	The strong rules that can be mined from D with given MST and MCT
R_S	R_S denote a set of sensitive rules that need to be hidden, and $R_S \subset R$
R_N	R_N is the set of non-sensitive rules, and $R_N \subset R$. $R = R_N \cup R_S$
R'	R' denotes the strong rules mined from the sanitized database D' with the same MST and MCT
S-N-H	The sensitive rules which still can be mined out in D' $\text{S-N-H} = \{r \in R_s \| r \in R'\}$
N-S-L	The non-sensitive rules which are falsely hidden and lost in D' $\text{N-SL} = \{r \in R_N \| r \notin R'\}$
S-F-G	The spurious rules falsely generated in D' $\text{S-F-G} = \{r \in R' \| r \notin R\}$

R_S is the set of sensitive rules that need to be hidden. R_N is the set of non-sensitive rules. $R_N \cup R_S = R$. The hiding problem is to transform D into a sanitized database D' such that only the rules set R_N can be mined from D'. R' denote the strong rules mined from sanitized database D' with the same MST and MCT. There are three possible side effects after transforming D into D'. They are S-N-H (Sensitive rules Not Hidden), N-S-L (Non-Sensitive rules Lost) and S-F-G (Spurious rules Falsely Generated), as indicated in Table 1.

We can formulate the sensitive rules hiding task as a multi-objective optimization problem as following:

Objective function:

$$\text{Minimize } \vec{f} = [f_1, f_2, f_3] = [|\text{S-N-H}|, |\text{N-S-L}|, |\text{S-F-G}|]$$

If the integer encoding is adopted in the EMO algorithm, the search/decision space is $\{1, 2, \ldots, n\}^{k}$. Where n is database size and the list 1, 2, …, n are the *ID*s of transactions in database. $n = |D|$. Parameter k is the number of transactions to be modified. We need to select k ones from n transactions to modify. However, if the binary encoding is used, the search space will be $2^{|D|}$. It is less than search space based on the integer encoding. Actually, only the transactions which support sensitive rules need to handled, so the search space can be reduced further.

In theory, the objective space is $\{0, 1, \ldots, |R_S|\} \times \{0, 1, \ldots, |R_N|\} \times \{0, 1, \ldots, |R'|\}$. In practice, compared with the tremendous decision space, the objective space is very sparse because the number of lost non-sensitive rules or spurious rules is far less than the theoretical maximum, i.e., |N-S-L| << $|R_N|$ and |S-F-G| << $|R'|$.

3 Sensitive Rules Hiding Method Based on EMO

In this section, we describe in detail the rule hiding approach based on binary encoded evolutionary multi-objective optimization algorithm. The model we adopted to hide sensitive rules is to modify a subset of transactions by removing items. An improved version of Apriori algorithm [12–15] is used to find all frequent itemsets and association rules under given *MST* and *MCT*.

We make the following assumptions for the rule hiding problem.

(1) We only consider the knowledge in the form of association rules which can be mined form the database. The knowledge of other kinds is not considered here.
(2) The proposed method operates on the binary or categorical dataset. In a binary dataset, the association rules contained in it do not indicate the number of items; they simply show the presence.
(3) When the rule's support is less than *MST* or the rule's confidence is less than *MCT*, we think this rule to be hidden. In other word, if a strong rule becomes not strong after sanitization, we think it as hidden.
(4) The sensitive rules are specified according to user's preference, organization enactments or policies, business interest conflicts between different units and etc.

3.1 Hiding Rules by Deleting Items

The supports or confidences of a sensitive rule can be reduced by removing some items from the transactions which support the sensitive rule. The removed item is the one with highest support/frequency corresponding to the consequent part of the sensitive rule. However, the modification of transaction may affect other rules' support or confidence when these rules are supported by the same transaction and share the removed item with the sensitive rule. The challenge behind this model is how to find subset of transactions to modify which can hide sensitive rules while incur minimal side effects. This is the place where the EMO algorithm can exhibit its strength.

3.2 The Architectural of the Proposed Method

The proposed algorithm architecture is based on the PISA platform [11]. In the PISA frame, the components of an EMO algorithm are divided into two parts: the variation part and the selector part. The variation part includes problem representation (chromosome encoding and objective function calculation) together with the variation operators and population initialization. Whereas, the fitness assignment, density estimation or diversity maintenance, environmental selection, parent selection and archive mechanism are combined as the selector part. The two parts communicate via file system. The architecture diagram is showed in Fig. 1. In the proposed method, the selector part is realized with the NSGA II algorithm [5]. The NSGA II uses the non-dominated environmental selection and the "crowding distance" density estimation strategy to rank the population and select the candidates for the next generation.

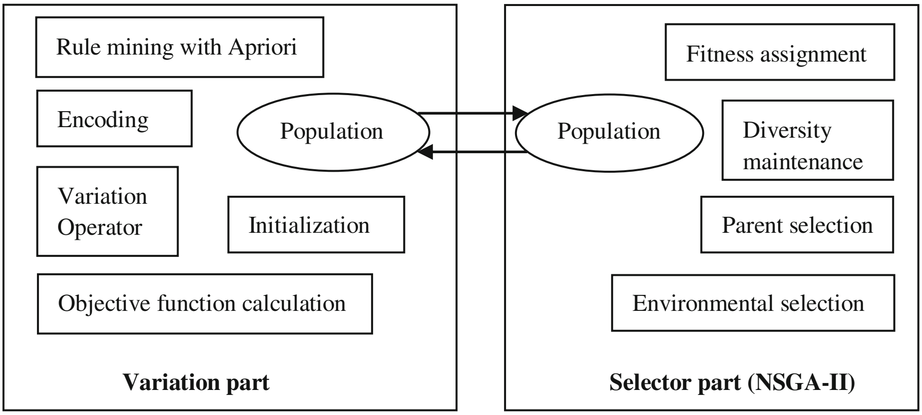


Fig. 1. The architectural diagram of the proposed method

3.3 The Binary Encoded Scheme for EMO

The chromosome encoding used in EMO algorithm is binary encoding. Every bit in the chromosome corresponds to a transaction. The bit with value 1 denotes that the corresponding transaction is selected to modify. The bit with value 0 indicates the corresponding transaction is not selected. If all the transactions in database are encoded based on this binary mechanism, the length of the chromosome will be very long and it may bring the problem of huge search space. Thus the transactions which support any sensitive rules are retrieved and filtered in advance. The bits in the chromosome only correspond to the supporting transactions. By this way, the size of search space is cut down greatly. We only need to find suitable candidates within supporting transactions.

Two different strategies can be used to map the transaction to binary bit in the chromosome. One way is to combine all the supporting transactions and map each one to a binary bit in the chromosome. The other way is slightly more complex but it can improve the search efficiency. Assuming that there are *n* sensitive rules, the chromosome is divided into *n* segments. Each segment in the chromosome corresponds to a sensitive rule. The length of the i^{th} segment is the number of transactions which support the i^{th} sensitive rule. The j^{th} bit in the i^{th} segment corresponds to the j^{th} supporting transaction for the i^{th} sensitive rule. Here, $1 \le i \le |R_S|$ and $1 \le j \le Supp\_N_i$. $Supp\_N_i$ is the number of transactions which support the i^{th} sensitive rule. In the second encoding way, a transaction can map to two or more bits in different chromosome segments since it might support several sensitive rules simultaneously. We adopted the second way in the proposed method.

3.4 How Many Transactions We Need Modify to Hide All Sensitive Rules?

One aim of data sanitization is to hide rules by causing side effects as less as possible. Fewer transactions to be modified mean fewer possible side effects. Therefore, it is an important issue to decide how many transactions at least need to be modified to hide all sensitive rules. For binary encoding, it is relevant to how many bits to be set as 1 to hide all rules. A fixed number cannot meet the variety of data source, sensitive rules and parameter settings. The better choice is to determine it dynamically. One sensitive rule can be hidden by reducing its support below *MST* or its confidence below *MCT*. Thus, the following properties can be deduced.

Property 1. Let Σ_{XUY} be the set of all transactions which support the sensitive rule $X \rightarrow Y$. In order to decrease the confidence of the rule below *MCT*, the minimal number of transactions which need to be modified in Σ_{XUY} is:

$$\mathrm{NUM}_1 = \lceil (Supp(X \cup Y) - Supp(X)^{*}MCT)^{*}|D| \rceil + 1 \tag{1}$$

Proof. Removing one item from the transaction in Σ_{XUY} which corresponds to the consequent part will decrease the support of the rule $X \rightarrow Y$ by 1. Assume θ is the minimal number of transactions which need to be removed in Σ_{XUY} in order to reduce the confidence of the rule below *MCT*. Then we have:

$$\begin{aligned} &(Supp(X \cup Y)^{*}|D| - \theta)/(Supp(X)^{*}|D|) < MCT \\ &\rightarrow Supp(X \cup Y)^{*}|D| - Supp(X)^{*}|D|^{*}MCT < \theta \end{aligned}$$

Because θ is an integer and θ is the minimum number which is greater than $Supp(X \cup Y)$ $^{*}|D|$ – $Supp(X)^{*}|D|^{*}MCT$, we can get:

$$\begin{aligned} &\theta > Supp(X \cup Y)^{*}|D| - Supp(X)^{*}|D|^{*}MCT \\ &\rightarrow \theta = \lceil (Supp(X \cup Y) - Supp(X)^{*}MCT)^{*}|D| \rceil + 1 \end{aligned} \qquad \square$$

Property 2. Let Σ_{XUY} be the set of all transactions which support the sensitive rule $X \rightarrow Y$. In order to decrease the support of the generating itemset for $X \rightarrow Y$ below *MST*, the minimal number of transactions which need to be modified in Σ_{XUY} is:

$$\text{NUM}_2 = \lceil (Supp(X \cup Y) - MST)^* |D| \rceil + 1 \tag{2}$$

Proof. Removing one item in a transaction belonging to Σ_{XUY} will decrease the support of the rule $X \rightarrow Y$ by 1. Assume θ is the minimal number of transactions which need to be removed in Σ_{XUY} in order to reduce the support of the rule below *MST*. Then we have:

$$(Supp(X \cup Y)^* |D| - \theta)/|D| < MST \rightarrow Supp(X \cup Y)^* |D| - MST^* |D| < \theta$$

Because θ is an integer and θ is the minimum number which is greater than *Supp* (*X*U*Y*) *|*D*| – *MST**|*D*|, we can get:

$$\theta > Supp(X \cup Y)^* |D| - MST^* |D| \rightarrow \theta = \lceil (Supp(X \cup Y) - MST)^* |D| \rceil + 1. \quad \square$$

Based on Property 1 and Property 2, we can infer the minimum number of transactions to be modified to hide the sensitive rule is:

$$\begin{aligned} &\text{Min}\{\text{NUM}_1, \text{NUM}_2\} \\ &\quad = \text{Min}\{\lceil (Supp(X \cup Y) - Supp(X)^* MCT)^* |D| \rceil + 1, \lceil (Supp(X \cup Y) - MST)^* |D| \rceil + 1\} \end{aligned} \tag{3}$$

The formula (3) gives the minimal number of transactions which need to be modified by removing items to hide the sensitive rule $X \rightarrow Y$.

3.5 Variation Operators and Initialization Strategy

Uniform crossover and independent bit mutation are utilized in the evolution process. Since the binary bits in the chromosome only correspond to the transactions which support sensitive rules, the offspring chromosome still holds the same length and its binary bits are also only relevant to supporting transactions.

In order to improve the quality of solutions and accelerate convergence, we may specify the minimal number of bits with value 1 for each segment of the chromosomes to hide sensitive rules in the initial population. We denote the minimal number of bits with value 1 in the i^{th} segment as NUM_{i}, corresponding to the i^{th} sensitive rule. Here $1 \leq i \leq |R_{\text{S}}|$. This minimal number is determined according to the formula (3). For each solution in the initial generation, the i^{th} chromosome segment contains the NUM_{i} bits with value 1 and other bits hold the value 0. These bits with value 1 are spread randomly in each segment but the total number of 1-bits should be NUM_{i} for the i^{th} segment. The initialization strategy can ensure that each individual in the first generation is a feasible solution to hide all sensitive rules. The difference within these individuals lies in their choice of bit locations with value 1 and accordingly may bring different side effects. The heuristic information on transaction length is added into one solution. For the i^{th} segment of this solution, the corresponding supporting transactions are sorted by their lengths and the shortest NUM_{i} transactions are selected to set as 1 in the relative bits.

However, the uniform crossover or independent mutation operators will break the rule in following generations and bring more or less bits with value 1 in the each segment of the offspring chromosome. In other word, the newly generated offspring individual can't ensure all sensitive rules to be hidden, but it might hide some (not all) sensitive rules with fewer side effects on non-sensitive lost rule or ghost rules. Thus the diversity is maintained and in the final outcome user may have more choices to modify the database.

3.6 The Procedure of Sensitive Rules Hiding

Algorithm 1 indicates the general procedure for hiding sensitive rules based on EMO. First, an improved Apriori algorithm is used to mine out frequent itemset and relevant strong association rules. When the association rules are obtained and the user has specified the sensitive ones from them, the hiding process begins. Transactions which support sensitive rules are filtered out so that the search is restricted to the subset of database which is relevant to sensitive rules. Then the initial population with μ individuals is generated and their objective values are evaluated. Then, the algorithm enters into the evolution process.

In each generation of evolution, λ new individuals are generated by variation operators. Then these offsprings are merged with μ archive individuals. The "fast non-dominated sort" algorithm in NSGA II [5] is applied in order to select out Pareto optimal ones from $(\mu + \lambda)$ solutions. Since different transactions subset selection may bring different side effects, the NSGA II algorithm is used to find candidates to modify which can hide sensitive rules with minimal side effects. Thus it is critical how to evaluate solutions to guide the search.

Algorithm 1. The general procedure of rules hiding based on EMO

INPUT: The original database *D*, *MST*, *MCT*, the sensitive rules.
OUTPUT: Sanitized database *D*' in which sensitive rules cannot be mined out.

```
BEGIN
    Find out frequent item sets and strong association rules in the original database D
      using improved Apriori algorithm
    Filter out supporting transactions for each sensitive rule.
    Determine the length of each segment in the chromosome and
      the minimal number of transactions to modify to hide each sensitive rule.
    $P_0 \leftarrow$ Generate the initial population, $t \leftarrow 0$.
    For each individual $x$ in $P_0$
        Evaluate the value of each optimization objective.
    Repeat
        $Q_t \leftarrow$ Generate($P_t$) with variation operators
        For each individual $x$ in $Q_t$
                Evaluate each objective's value.
        $P_t \leftarrow (P_t \cup Q_t)$.
        Fast-nondominated-sort ($P_t$ ) and estimate each solution's density in $P_t$
        Fitness assignment and sorting in descending order
        Environmental selection: Reduce $P_t$ from $(\mu+\lambda) \rightarrow \mu$.
        Parent selection
        $t \leftarrow t+1$.
    Until the max generation is reached.
    Choose the preferred solution from Pareto optimal set.
    Modify the database using the selected preferred solution.
END
```

3.7 How to Evaluate Each Solution?

One of the challenging issues to use evolutionary algorithm to perform association rule hiding is how to evaluate the solution's objective functions efficiently. This is the most time-consuming part in the algorithm. The optimization goals/objective functions are three side effects, i.e., |S-N-H|, |N-S-L| and |S-F-G|. There are the following two options to calculate these three side effects:

(1) Modify corresponding transactions in database according to the solution's genotype, and mine out new rules from the modified database. Then compare the original rule sets with new rules sets in order to determine the three side effects. This way is very time consuming and it is infeasible in time since we need to do it for each solution in each generation.
(2) Copy rules set *R* mined from the original database to *R*'. According to the solution's genotype, determined the transaction to be modified and the item to be

removed, then update the rules set copy. For each rule in *R*, compare the original support and confidence with new ones to determine the side effects (objective functions).

The details for the second method in indicated in Algorithm 2.

```
Algorithm 2.  Eval (Individual x)

INPUT: The chromosome x, which includes |$R_S$| segments.
OUTPUT: The objective vector < obj1, obj2, obj3 >

BEGIN
Copy the rules set R to another set R'; Copy thin database D to D'.
For each segment $x_i$ in x
    For each gene bit g in $x_i$
        if (1=g)
        {
            Identify the corresponding transaction t.
            Choose the item with highest support and corresponding to
            the consequent part of the $i^{th}$ sensitive rule to remove from t.
            The database D' is updated.
            Count the number of items which are modified.
            For each rule r in R'
                Update support and confidence of r according to t.
        }
    // The new support and new confidence of each rule are got.
    // We denote them as new_Supp(r) and new_Conf(r) respectively.
For each rule r in $R_S$
        if (new_Supp (r) $\geq$ MST and new_Conf(r) $\geq$ MCT)
            |S-N-H| = |S-N-H| + 1
For each rule r in R - $R_S$
    {
        if ( Supp(r) $\geq$ MST and Conf(r) $\geq$ MCT ) and
           ( new_Supp(r) < MST or new_Conf(r) < MCT )
            |N-S-L| = |N-S-L| + 1

        if ( Supp(r) < MST or Conf(r) < MCT ) and
           ( new_Supp(r) $\geq$ MST and new_Conf(r) $\geq$ MCT )
            |S-F-G| = |S-F-G| + 1
    }
obj1 = |S-N-H|;  obj2 = |N-S-L|;  obj3 = |S-F-G|

END
```

3.8 Implemental Discussion

(1) In order to find out the spurious rules, we adopted a smaller confidence threshold to mine association rules than *MCT*. Because the proposed method hides sensitive rules by removing items, it is impossible to increase the support of any rule.

However, it may reduce the support of the antecedent part of pre-strong rules. The pre-strong rule represent the rules with support not less than *MST* but with confidence less than *MCT*. When the support of the antecedent of a pre-strong rule decreases but the union support of the whole rule remain same, the confidence of the rule will rise. The pre-strong rules may become strong when its confidence is equal or greater than *MCT*.

(2) In order to save memory space and improve the running efficiency, the original database was transformed into the so-called "thin" database. The algorithm only operated on the "thin" database. In the "thin" database, each transaction only contains frequent items whose support is greater than *MST*. The non-frequent items are ruled out since they are useless to the algorithm. It utilized the "downward closeness" property of frequent itemsets.

(3) The special data structures were adopted to improve the performance of objective function calculation. For the rules with two items, we used the two-dimensional array to store the supports of the rules. The use of array brings the benefit that the data can be accessed directly in a mapping way. The algorithm need not to retrieve the whole rules set to find the one for updating its support, as indicated on the line 9 in Algorithm 2. For rules which hold more than two items, the Trie-tree data structure [14, 15] was used to store and retrieve the corresponding generating itemsets. It can give a performance improvement for the "large" rules with many items.

4 Performance Evaluations

We tested the proposed algorithm on three well-known real databases: mushroom BMS-WebView-1 and BMS-WebView-2. These datasets exhibit varying characteristics with respect to the number of transactions and items that they contain, as well as with respect to the average transaction length. The experiments were carried out on an Intel Core(TM) i3 CPU with 2.53 GHz processor and with 2 GB of main memory. The proposed algorithm was implemented in C++ based on the PISA platform [11]. The experiment results were measured according to three side effects: (|S-N-H|, |N-S-L|, |S-F-G|).

The algorithm ran with maximal evolution generation as 200 and the population size as 20 (10 archive solutions and 10 offsprings). For the variation part, the crossover probability was 0.95. The mutation probability was 0.1 and the bit turn probability is 0.2. For NSGA II, the tournament size was 2. The Algorithms *1.a*, *1.b*, *2.a* and *2.b* proposed in [1] were used for comparison. Note that the Algorithm 1.a hides rules by adding items. All other algorithms hide rules by removing items.

Table 2 shows the experiment results. Five sensitive rules were selected randomly for each dataset to perform hiding task. For each dataset, the same sensitive rules set were used. As the Table 2 show, the values of |S-N-H| are zero in all experiments. It demonstrates all algorithms can effectively hide sensitive rules completely. Lowered *MCT* values generate more numbers of rules and it also increase the possibility of non-sensitive rules to be affected by the hiding process. This is demonstrated by the fact that there were more non-sensitive rules missing when the *MCT* was decreased. We also can notice that in most cases the proposed method can hide rules with fewer side effects.

Table 2. Side effects with different *MCT*s on three datasets

Dataset	MCT	|R|	Side effects: (|S-N-H|, |N-S-L|, |S-F-G|)				
			EMO-based (binary encoded)	1.a	1.b	2.a	2.b
Mushroom (*MST* = 0.05)	0.6	849	(0,7,1)(1,7,0)	(0,7,0)	(0,16,1)	(0,9,1)	(0,9,1)
	0.7	678	(0,10,0)(3,9,0)	(0,10,0)	(0,13,0)	(0,12,0)	(0,12,0)
	0.8	560	(0,5,0) (1,4,0)	(0,10,0)	(0,5,0)	(0,5,0)	(0,6,0)
	0.9	461	(0,3,0)	(0,12,0)	(0,4,0)	(0,4,0)	(0,10,0)
BMS-1 (*MST* = 0.001)	0.3	325	(0,4,0)	(0,7,0)	(0,4,0)	(0,4,0)	(0,11,0)
	0.4	131	(0,1,0)	(0,4,0)	(0,1,0)	(0,1,0)	(0,5,0)
	0.5	34	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,5,0)
	0.6	11	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,1,0)
BMS-2 (*MST* = 0.002)	0.3	482	(3,9,0) (2,10,0) (1,11,0) (0,12,0)	(0,9,0)	(0,18,0)	(0,12,0)	(0,13,0)
	0.4	283	(2,7,0)(1,8,0) (0,9,0)	(0,8,0)	(0,13,0)	(0,9,0)	(0,10,0)
	0.5	112	(0,6,0)	(0,8,0)	(0,6,1)	(0,6,1)	(0,7,1)
	0.6	29	(0,2,0)	(0,2,0)	(0,2,0)	(0,2,0)	(0,3,0)

The evolutionary multi-objective optimization algorithm can produce multiple Pareto optimal solutions on a single run, and we need make further selection from them using preference information. However, on the association rule hiding problem, the experiment outcome only consisted of few different solutions in the final result. This special phenomenon is mainly caused by the sparseness of the objective space. Compared with the tremendous decision search space, the objective space is very sparse and discrete. Few different solutions exist in the objective space for the above three datasets. In addition, the ideal Pareto front for the problem of rule hiding is the single vector (0, 0, 0). Although it often cannot be achieved, the EMO algorithm endeavored to approximate this ideal solution. Thus it is not surprise that the outcome often included one or several few different solutions.

5 Conclusions

In this paper, we devised a new association rule hiding approach based on binary encoded evolutionary multi-objective optimization. Taking the rules hiding as a multi-objective combination optimization problem, the goal is to find optimal solutions which can minimize the number of sensitive rules not hidden, the number of missing non-sensitive rules and the number of spurious rules simultaneously. The algorithm modifies the database by deleting some items so as to decrease the support or confidence of sensitive rules below specified thresholds. The non-dominated ranking mechanism of NSGA-II is utilized to drive the evolution forward. The particularity of the adopted multi-objective model is the objective space is very sparse, and the challenge lies in the very huge decision search space.

Experiment results on three real data sets showed that the proposed approach can effectively hide all sensitive rules with few side effects. We can notice that the side effects mainly occur on the non-sensitive rules lost. It is an important topic to devise the variation part to efficiently cope with the tremendous decision search space. In addition, the data distortion/accuracy degree is not considered here, the optimization model will be improved to consider it in the future work.

References

1. Verykios, V.S., Elmagarmid, A.K., et al.: Association rule hiding. IEEE Trans. Knowl. Data Eng. **16**(4), 434–447 (2004)
2. Atallah, M.B.E., Elmagarmid, A., Ibrahim, M., Verykios, V.S.: Disclosure limitation of sensitive rules. In: Proceedings of IEEE Workshop on Knowledge and Data Engineering Exchange, Chicago, IL, pp. 45–52 (1999)
3. Dasseni, E., Verykios, V.S., Elmagarmid, A.K., Bertino, E.: Hiding association rules by using confidence and support. In: Proceedings of the 4th International Workshop on Information Hiding, pp. 369–383 (2001)
4. Wu, Y.H., Chiang, C.C., Chen, A.L.P.: Hiding sensitive association rules with limited side effects. IEEE Trans. Knowl. Data Eng. **19**(1), 29–42 (2007)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
6. Amiri, A.: Dare to share: Protecting sensitive knowledge with data sanitization. Decis. Support Syst. **43**(1), 181–191 (2007)
7. Verykios, V.S.: Association rule hiding methods. Wiley Interdisc. Rev. Data Min. Knowl. Disc. **3**(1), 28–36 (2013)
8. Sun, X., Yu, P.S.: A border-based approach for hiding sensitive frequent itemsets. In: Proceedings of Fifth IEEE International Conference on Data Mining (ICDM '05), pp. 426–433, (2005)
9. Divanis, A.G., Verykios, V.S.: Exact knowledge hiding through database extension. IEEE Trans. Knowl. Data Eng. **21**(5), 699–713 (2009)
10. Hong, T.P., Lin, C.W., Yang, K.T., Wang, S.L.: Using TF-IDF to hide sensitive itemsets. Appl. Intell. **38**(4), 502–510 (2013)
11. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA – a platform and programming language independent interface for search algorithms. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 494–508. Springer, Heidelberg (2003)
12. Agrawal, R., Imielinski, T., Sawmi, A.: Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD Conference on Management of Data (SIGMOD), pp. 207–216 (1993)
13. Agrawal, R., Srikant, R.: Fast algorithm for mining association rules. In: Proceedings of the International Conference on Very Large Data Bases (VLDB), pp. 487–499 (1994)
14. Bodon, F.: Surprising results of trie-based FIM algorithms. In: Proceedings of IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI '04), Brighton, UK (2004)
15. Bodon, F.: A fast APRIORI implementation. In: IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI '03), Melbourne, Florida, USA (2003)

Detecting the Data Group Most Prone to a Specific Disguise Value

Wen-Yang Lin(✉) and Wen-Yu Feng

Department of Computer Science and Information Engineering,
National University of Kaohsiung, Kaohsiung 811, Taiwan
wylin@nuk.edu.tw, m1005504@mail.nuk.edu.tw

Abstract. Disguised missing data, an emerging data quality problem coined by Pearson in 2006, is a special kind of missing data that refers to values not exactly missing in the data entries, but cannot reflect the fact and so may lead to severe bias on analysis results. In this paper, we present a novel problem of detecting disguised missing data, i.e., finding out the data group most prone to a specific disguise value. We show that this problem can be formalized as an optimization problem and so a genetic-algorithms-based method is proposed to handle this problem. According to preliminary experimental results conducted on real datasets, our method can discover the same optimal data groups obtained by exhaustive method. A further evaluation on the FDA adverse drug event reporting dataset shows that our method yields similar results concluded by manual examinations performed by experienced analyzers.

Keywords: Data cleansing · Disguised missing data · Genetic algorithms · Missing at random · Unbiased sampling

1 Introduction

Disguised missing data, just as the name implies, is a kind of missing data but they usually disguise themselves as fake values which cannot reflect the true data in real world. For example, in many online applications that require customers filling their information, some customers usually decline to disclose their private information so that they fill the field with wrong values. Namely, the missing values are not explicitly represented as such, but become potential factors that severely reduce the data quality of data analysis. In general, the problem of detecting disguised missing data is difficult because the disguised data appear as valid values for specific attributes and normally incur no data integrity problem.

Since that missing data can be classified into three types according to how missing entries were distributed in the whole dataset [6], it is reasonable that disguised missing data can also be classified accordingly, i.e., missing completely at random (MCAR), missing at random (MAR), and not missing at random (NMAR). MCAR refers to the case that the missing data is randomly distributed on the whole dataset, MAR names the situation that missing observations are dependent on other non-missing values, while NMAR means that the missing observations are dependent on the missing values themselves.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 100–111, 2014.
DOI: 10.1007/978-3-319-13186-3_10

Contemporary approaches for detecting disguised missing data fall into two categories. The first one refers to semi-manual approaches relied on some domain knowledge [9]; the second one refers to heuristic ways to detect disguised missing data automatically [1, 4]. The first category uses some statistic techniques and requires domain knowledge to assist the analysis. While the second category can automatically detect the most suspected values being used as a disguise, they only can handle the first type of disguised missing data, that is, the disguise value is randomly distributed in the whole dataset. To the best of our knowledge, no study has been conducted to devise automatic methods to detect the other two types of disguised missing data.

In this paper, we present a novel problem belonging to the second type of disguised missing data, that is, given a suspect or known disguised value, figure out which data group is most likely to raise the disguised missing value. We formalize this problem as an optimization problem by adapting the concept of previously developed embedded unbiased sample heuristic and propose a genetic-algorithms (GA) based algorithm for detecting the data group which is most prone to a specific disguised missing value. According to our experimental results, our method can generate the same solutions as found by the exhaustive method. A further evaluation on the FDA adverse drug event reporting dataset shows that our method yield similar results concluded in [9], which is relied on manual examinations performed by experienced analyzers.

2 Related Work

Disguised missing data was coined by Pearson in 2006 [9]. In the study, Pearson described the source of disguised missing data, and illustrated the influence of disguised missing data on simple statistics, hypothesis tests, correlations and regression models, and classification trees, then discussed if the record should be ignored or not. The disguise value is manually identified by finding unusual values or patterns in the dataset, utilizing some domain knowledge or through statistics-based univariate outlier or distributional anomalies detection methods, such as quantile-quantile (Q-Q) plot.

The study conducted by Hua and Pei [4] was the first work that proposed an automatic method for detecting disguised missing data, called EUS heuristic, which is based on the concept of embedded unbiased sampling. This method finds the unbiased sample based on the correlation-based sample quality score (CBSQS), and outputs the k most suspect disguised missing values on a given attribute. The method is based on the assumption that missing values are distributed randomly through the dataset and so it aims at detecting the disguised missing data of the first type. Although the authors claim their method can also be applied to detecting the second type of disguise missing value, no mechanism has been developed to locate the most data group.

In [1, 2], Belen modified the EUS heuristic method, replacing the evaluation of unbiased sample by a chi-square two sample test. The chi-square two sample test can check whether two samples come from the same distribution without the need to specify what the common distribution is. Motivation of this approach is to relieve the deficiency caused by data dependency of attributes values. Experimental results show the proposed measure can alleviate the dominating effect caused by attribute dependency but it improves not too much for datasets having no attribute dependency.

Natarajan et al. [8] suggested using an association rule based framework for detecting disguised missing data in large dataset. Their method requires a training set without disguised missing data to generate association rules of attribute values with 100 % confidence, and then use this knowledge to determine disguise missing values. Unfortunately, the training set is not available in most real applications. Furthermore, the effectiveness of the suggested framework is questionable since no experiments were conducted.

3 Problem Description

3.1 Preliminary

Since our approach adapts some concept of the EUS heuristic proposed by Hua and Pei [4], in this subsection we first introduce the main idea behind this heuristic.

The EUS heuristic is based on two basic assumptions. Firstly, for an attribute, there often exist only a small number of disguises that are frequently used by the disguised missing data. Those values are called the *frequently used disguises*. Secondly, the disguised tuples are randomly distributed in the whole dataset.

Let T be the truth table and $\tilde{T}$ be the recorded table. $T_{A\,=\,v}$ is call the *projected database* of v that all the tuples in $T_{A\,=\,v}$ contain value v on attribute A. For simplicity, we denote $T_{A\,=\,v}$ as T_v.

Definition 1 (EUS heuristic [4]). *If v is frequently used as a disguise value on attribute A, then there exists a large subset $S_v \subseteq \tilde{T}_v$ such that S_v is an unbiased sample of $\tilde{T}$ except for attribute A.*

According to the *EUS heuristic*, S_v is an unbiased sample of $\tilde{T}_v$. The larger S_v, the more frequently v is used as a disguise. So a value with the largest unbiased sample is the most possible disguise value. To measure whether a subset S_v is an unbiased sample of $\tilde{T}_v$, Hua and Pei proposed the correlation-based sample quality score $\phi(\tilde{T}_v, S_v)$, CBSQS in short, which is defined as follows:

$$\phi(\tilde{T}_v, S_v) = \sum_{P_{S_v}(v_i,v_j)\,>\,0} \frac{P_{\tilde{T}_v}(v_i, v_j)}{1 + \left|Corr_{\tilde{T}_v}(v_i, v_j) - Corr_{S_v}(v_i, v_j)\right|^{q'}}, \tag{1}$$

where $Corr_{\tilde{T}_v}(v_i, v_j)$ and $Corr_{S_v}(v_i, v_j)$ denote correlations of v_i and v_j on $\tilde{T}_v$ and S_v, respectively, $P_{\tilde{T}_v}(v_i, v_j)$ represents the probability of value pair (v_i, v_j) on $\tilde{T}_v$, and q' imitates the Minkowski distances. The score obtained by CBSQS is a non-negative number; the higher the score of subset S_v, the better S_v is an unbiased sample of $\tilde{T}_v$.

3.2 Formal Definition

In this study, we focus on the disguised missing data that is missing at random. That is, a disguise value is randomly distributed in a specific subset of the whole database. For example, when customers are filling an application form on the internet, they may not want to reveal their private information such as birth date, age, country, etc. A man, for example, whose "*Birth date*" is "*February 29th*", after entering "*February*" to "*Month*", intends not to disclose his true information on "*Birth date*". So he chooses the default value, says "1" for "*Day*". Similarly, there may also be some other customers born on "*February*" choosing "*February 1st*" as a disguise. As a result, "*February 1st*" becomes a disguise value in the subset containing "*February*" on attribute "*Month*" though it is usually not a disguise value on the whole dataset, which is a typical scenario of disguised data missing at random. In this subsection, we show this problem can be formalized as an optimization problem.

Following the notation used in [4], we use $\tilde{T}$ to denote the recorded table of a truth table T with attributes $A = \{A_1, A_2, \ldots, A_n\}$, and $Dom(A_i)$ the set of values for attribute A_i, $1 \le i \le n$. Given a suspected disguise value v, for $v \in Dom(D)$ and $D \in A$, we like to discover if v is indeed a disguise value, the data group of $\tilde{T}$ that is most prone to using v as a disguise value. To facilitate the discussion, we first formalize the term *data group*.

Definition 2. *A data group* $(G_1 = g_1, G_2 = g_2, \ldots, G_p = g_p)$ *defined on a attribute subset* $\{G_1, G_2, \ldots, G_p\} \subseteq A$, *identifies the projection* of $\tilde{T}$ *on* $G_1 = g_1$, $G_2 = g_2, \ldots, G_p = g_p$. *That is, the set of tuples in group* $(G_1 = g_1, G_2 = g_2, \ldots, G_p = g_p)$ *all have the same values on attributes* $G_1, G_2, \ldots, G_p$. *Hereafter, as it is clear from the context, we use* $(g_1, g_2, \ldots, g_p)$ *instead of* $(G_1 = g_1, G_2 = g_2, \ldots, G_p = g_p)$.

Now let $\mathcal{G}$ denote the set of all data groups induced by the attribute set $A - \{D\}$. Note that we need at least one attribute other than the grouping attributes as well as the disguise attribute D to perform the EUS procedure. It is noteworthy that the empty group means no projection is performed on the original table $\tilde{T}$. So this case corresponds to the discovery of the maximal embedded unbiased sample on $\tilde{T}_v$. In this context, the problem discussed in [4] can be regarded as a special case of our problem.

Based on the concept of maximal embedded unbiased sample, we can formalize the problem of detecting the data group most prone to the specific disguise value v as finding the best data group $g*$ in $\mathcal{G}$ that maximizes Eq. (1). Since the searching of the maximal embedded unbiased sample is performed on the projection of $\tilde{T}$ on v associated with group g, i.e., $\tilde{T}_{v,g}$, instead of $\tilde{T}_v$, the problem is now formalized as

$$g* = \arg\max_{g \in \mathcal{G}} \left\{ \max_{U \subseteq \tilde{T}_{v,g}} \left\{ \frac{|U|}{\left|\tilde{T}_{v,g}\right|} \phi(\tilde{T}_g, U) \right\} \right\}. \tag{2}$$

Recalling the formula of CBSQS in Eq. (1), the maximal value occurs when $\tilde{T}'$ is the same as $\tilde{T}$, making $Corr_{\tilde{T}}(v_i, v_j) = Corr_{\tilde{T}'}(v_i, v_j)$ and so the denominator equals to 1. The maximal value is $\sum_{P_{\tilde{T}'}(v_i, v_j) > 0} P_{\tilde{T}}(v_i, v_j)$, which is proportional to the cardinality of the table of concern. The larger the cardinality (number of value pairs) of table $\tilde{T}$, the larger this value is. In order to not favor larger projections of $\tilde{T}$ on smaller groups and so bias the results of detection, we normalize the CBSQS score by its maximal value, called normalized CBSQS, denoted as $\bar{\phi}(\tilde{T}, \tilde{T}')$.

$$\bar{\phi}(\tilde{T}, \tilde{T}') = \left(\sum_{P_{\tilde{T}'}(v_i, v_j) > 0} \frac{P_{\tilde{T}}(v_i, v_j)}{1 + \left| Corr_{\tilde{T}}(v_i, v_j) - Corr_{\tilde{T}'}(v_i, v_j) \right|^{q'}} \right) \Big/ \sum_{P_{\tilde{T}'}(v_i, v_j) > 0} P_{\tilde{T}}(v_i, v_j) \tag{3}$$

Similarly, we introduce the normalize DV-score of v in $\tilde{T}$, denoted as $ndv(v, \tilde{T})$.

$$ndv(v, \tilde{T}) = \max_{U \subseteq \tilde{T}_v} \left\{ \frac{|U|}{|\tilde{T}_v|} \bar{\phi}(\tilde{T}, U) \right\} \tag{4}$$

Therefore, the problem is to find the best group g^* that maximizes the normalized DV-score $ndv(v, \tilde{T}_g)$, i.e.,

$$g^* = \arg\max_{g \in \mathcal{G}} \left\{ ndv(v, \tilde{T}_g) \right\}, \tag{5}$$

The complexity of finding g^* is immense. Let m_i be the cardinality of attribute A_i in $\tilde{T}$, $1 \leq i \leq n$. Without loss of generality, we choose A_1 as the suspected attribute D. Each attribute A_j, $2 \leq j \leq n$, can take either one of m_j different values if being involved in forming the data group or take the empty value if not being involved, leading to at most $(m_2 + 1) \times (m_3 + 1) \times \ldots \times (m_n + 1)$ different data groups. Since at least one attribute has to be excluded in forming the data group, we have to discount all the cases that all attributes are involved in forming the data group. Then, the number of all possible data groups induced by the set $\{A_2, A_3, \ldots, A_n\}$ is

$$\prod_{j=2}^{n} (m_j + 1) - \prod_{j=2}^{n} m_j, \tag{6}$$

which is at least the order of m^{n-2}, for $m = \min\{m_1, m_2, \ldots, m_n\}$.

Example 1. Let us consider Table 1. Suppose we choose "*male*" on "*Gender*" as the suspected disguise value v. Then the number of data groups induced by attributes "*Martial Status*", "*Literacy*", and "*Education*" is (|*Dom*(*Martial Status*)| + 1) × (|*Dom*(*Literacy*)| + 1) × (|*Dom*(*Education*)| + 1) − (|*Dom*(*Martial Status*)| × |*Dom*(*Literacy*)| × |*Dom*(*Education*)|) = $3^3 - 2^3 = 19$. Specifically, let us consider the group defined on *Martial Status* = "*married*". Table 2 shows the resulting projection $\tilde{T}_{married}$ on this group, wherein the shaded part corresponds to further projection on "*Male*", say $\tilde{T}_{male,married}$. According to Eq. (4), we have to find the maximal subset of $\tilde{T}_{male,married}$ that resembles (an unbiased sample of) $\tilde{T}_{married}$. This process continues for the projections defined on all other groups to determine the best data group g^*.

Table 1. An example dataset.

Gender	Marital Status	Literacy	Education
Male	Married	Literate	High school
Male	Single	Literate	High school
Male	Married	Illiterate	High school
Male	Single	Illiterate	High school
Male	Married	Literate	College
Male	Single	Literate	College
Male	Married	Illiterate	College
Male	Single	Illiterate	College
Male	Married	Literate	High school
Male	Single	Literate	High school
Female	Married	Illiterate	High school
Female	Single	Illiterate	High school
Female	Married	Literate	College
Female	Single	Literate	College
Female	Married	Illiterate	College
Female	Single	Illiterate	College
Male	Married	Literate	High school
Female	Single	Literate	College
Female	Married	Illiterate	College
Female	Single	Illiterate	High school

Table 2. The resulting projection of Table 1 on "*Married*".

Gender	Marital Status	Literacy	Education
Male	Married	Literate	High school
Male	Married	Illiterate	High school
Male	Married	Literate	College
Male	Married	Illiterate	College
Male	Married	Literate	High school
Male	Married	Literate	High school
Female	Married	Illiterate	High school
Female	Married	Literate	College
Female	Married	Illiterate	College
Female	Married	Illiterate	College

4 The Proposed GA-Based Detection Method

Our proposed GA-based method follows the framework of simple genetic algorithms (SGAs) [7]. Beginning with an initial population (group) of randomly generated chromosomes (solutions), SGAs choose parents and generate offspring using operations analogous to biological processes, i.e., crossover and mutation. All chromosomes are evaluated according to a fitness function; the higher fitness chromosomes are kept and the less ones are discarded in generating a new population to replace the old one. The whole process continues until a specific termination criterion is satisfied. In the end, the chromosome with the highest fitness value gives the solution.

4.1 Chromosome Representation

A chromosome representation is an encoding of a possible solution of the problem. Rather than adopting binary encoding used by SGAs, we encode each solution into a vector of non-repeated decimal integers. A non-zero integer indicates the corresponding attribute values used for forming the data group, while a zero value represents the attribute not included in forming the group but used for evaluating the degree of fitness using the EUS heuristic.

For example, consider a four attribute table T, whose attribute A_1 contains two values, A_2 three values, A_3 two values, and A_4 two values. So in total we have 9 different attribute values, which are mapped into integers of 1 to 9. If we choose v_{11} on A_1 and v_{32} on A_3 for grouping and leave A_2 and A_4 as fitness evaluation attributes, then the chromosome can be represented as a vector "1 0 7 0".

4.2 Evolutionary Operations

It is necessary to choose the parent chromosomes from the population before evolutionary operation, which is called a selection. According to the evolution principle, choosing the chromosomes with higher degree of fitness can generate better population. However, this approach may lose population diversity because of restricting the possible solutions. The population will converge too quickly and may not be able to find the optimal solution. In this work, we adopted the tournament selection method [7], which randomly chooses parent chromosomes from the current population and a random number r between 0 and 1 is generated and compared with a predefined value, usually set as 0.75. If r is less than the value, we choose the chromosome with higher fitness value; otherwise, we choose the chromosome with lower fitness. We also adopted the elitism principle [7], preserving the best chromosome into the new population.

The crossover operation is used to generate the offspring by exchanging the chromosome in two parents chosen from population. Our method adopts one-point crossover, which is one of the most common crossover operations. This operation works by first selecting a crossover point randomly, dividing the pair of parents by this point, and then exchanging the gene sequence to form the offspring.

Mutation operation is used to increase genetic diversity. In our method, the position for mutation is selected randomly. The value of the selected gene mutates in the following way. If the gene is zero, it is changed to a random non-zero integer. On the other hand, if the gene is non-zero, it changes to another integer including zero.

4.3 Fitness Function

Intuitively, we can adopt the normalized DV-score $ndv(v, \tilde{T}_g)$ described in Eq. (3) as the fitness function to measure the possibility that v is used as a disguise in the projection $\tilde{T}_g$ induced by the data group g represented by the chromosome. Note that $ndv(v, \tilde{T}_g)$ requires computing the normalized CBSQSs for each subset U of $\tilde{T}_{v,g}$, which consumes lots of computations proportional to the number of different attribute value pairs in U. Although it is not easy to reduce the complexity of the normalized CBSQS, we can simplify the denominator term to $C(k, 2)$, where k denotes the number of attributes in $\tilde{T}$ not serving as the disguise and grouping attributes.

Lemma 1. *Consider a subset U of the projected table* $\tilde{T}_{v,g}$. *We have*

$$\sum_{P_U(v_i,v_j)>0} P_{\tilde{T}_g}(v_i, v_j) \le C(k,2). \tag{7}$$

Proof. Since $U \subseteq \tilde{T}_{v,g}$, U has the same set of attributes as that in $\tilde{T}_{v,g}$. That is, the attribute cardinality of U is k. Without loss of generality, let us consider any two attributes of U, say A_p and A_q. Intuitively, the total probability of value pairs from any two attributes should be equal to 1. Therefore, we have

$$\sum_{\forall(v_i \in A_p, v_j \in A_q)} P_{\tilde{T}_g}(v_i, v_j) = 1. \tag{8}$$

Since U consists of k attributes, if we select two attributes from these k attributes once a time, then we obtain in total $C(k, 2)$ combinations. It follows that in $\tilde{T}_g$ the total probability of all value pairs from these k attributes is $C(k, 2)$. Note that not every value pair appearing in $\tilde{T}_g$ also appears in U, which completes the proof.

Let g be the corresponding data group encoded by a given chromosome χ. The fitness function for evaluating χ is defined using the following simplified normalized DV-score.

$$\begin{aligned} fitness(\chi) &= \max_{U \subseteq \tilde{T}_{v,g}} \left\{ \frac{|U|}{|\tilde{T}_{v,g}|} \bar{\phi}(\tilde{T}_g, U) \right\} \\ &= \max_{U \subseteq \tilde{T}_{v,g}} \left\{ \frac{|U|}{|\tilde{T}_{v,g}| \times C(k,2)} \left(\sum_{P_U(v_i,v_j)>0} \frac{P_{\tilde{T}_g}(v_i, v_j)}{1 + \left| Corr_U(v_i, v_j) - Corr_{\tilde{T}_g}(v_i, v_j) \right|} \right) \right\} \end{aligned} \tag{9}$$

4.4 Candidate Pruning

As shown in Sect. 3.2, the search space of candidate data groups is in the order of $O(m^{n-2})$, an exponential function of m and n. In order to avoid unnecessary exploration of the search space, we developed several optimization techniques to prune unqualified candidates. These optimizations include attribute-based pruning, value-based pruning, record-based pruning, and hierarchy-based pruning.

Optimization 1 (attribute-based pruning): Any data group with cardinality larger than $n - 2$ should be pruned, where n is the number of attributes in $\tilde{T}$. This is because EUS-based fitness function requires at least two nongrouping and disguise attributes to calculate the correlation between value pairs illustrated in Eq. (2). Our approach ensures all chromosomes generated in the initial population are qualified, and enforces this rule to the operation of crossover and mutation. Specifically, if any offspring generated by a crossover contains less than two zero genes, then the crossover is a failure mating, and so we discard the offspring and instead keep the parents to the next generation. Similarly, if there are exactly two zero genes in the chromosome undergone mutation, then the mutation operation will select a nonzero gene to change its value.

Optimization 2 (value-based pruning): Any data group resulting a projection $\tilde{T}_g$ containing only one value on the disguise attribute D should be excluded, no matter the value equals to the disguise value v or not. This is because in this case $\tilde{T}_{v,g}$ will be empty (if the value is not v) or equal to $\tilde{T}_g$ (if the value is v), both making the correlation computation meaningless. Our approach ensures the initial population excluding such kind of candidates and punishes any chromosomes generated after crossover or mutation operation that resulting only one value on attribute D by assigning these chromosomes an extremely small fitness.

Optimization 3 (record-based pruning): Any data group resulting in a projection $\tilde{T}_g$ containing a relatively small amount of records will be pruned. This is because a smaller subset tends to lose good representation of the original dataset. Therefore, we avoid creating candidates with this problem during generating the initial population, and also assign an extremely small fitness to any chromosome with this problem after the processes of crossover and mutation.

Optimization 4 (hierarchy-based pruning): This optimization prunes candidates by exploiting the hierarchy information existing between attribute values. Consider a group $g = (g_1, g_2, \ldots g_p)$. If there exist two values g_i and g_j, and g_i is a descendant of g_j in the value hierarchy, then g can be pruned and replaced by $g' = g - \{g_j\}$, i.e., $g' = (g_1, g_2, \ldots, g_{j-1}, g_{j+1}, g_p)$. This is because the resulting projections $\tilde{T}_g$ and $\tilde{T}'_g$ have exactly the same tuple values in every nongrouping attribute.

5 Experiments

We conducted experiments to evaluate the proposed GA-based method. Our experiments consist of two parts: The first part focuses on the performance of our method. The second part shows the correctness of the solution. To evaluate the efficiency and the correctness of our GA-based method, we compared it with the exhaustive method, simply evaluating all possible candidates in set $\mathcal{G}$ to find the best solution. We used the same two datasets considered in [4, 9], including the Pima Indians Diabetes dataset [10] and FDA Adverse Event Reporting System (FAERS) dataset [3].

All experiments were performed on a personal computer running the Microsoft Window 7, with Intel Core i7-2600 3.4 Ghz CPU, 8 GB main memory, and a 500 GB hard disk. All programs were coded in C#. The order q' in Eq. (3) was set to 1. The other parameter settings used in our method are *max generation* = 100, *population size* = 30, *crossover probability* = 0.75, and *mutation probability* = 0.033.

5.1 Evaluation on Execution Time

In this experiment we used the Pima Indians Diabetes dataset to evaluate our method with respect to the size of the dataset and compared it with the exhaustive method. This dataset consists of 768 records for females from the Pima Indian tribe, each composed of eight clinical predictor attributes, e.g., NPG (Number of times pregnant) BMI (Body mass index), along with the patient's diagnosis as diabetic or nondiabetic. In order to obtain a larger dataset, we duplicate the Pima Indians Diabetes dataset up to five times. The experimental result is shown in Fig. 1.

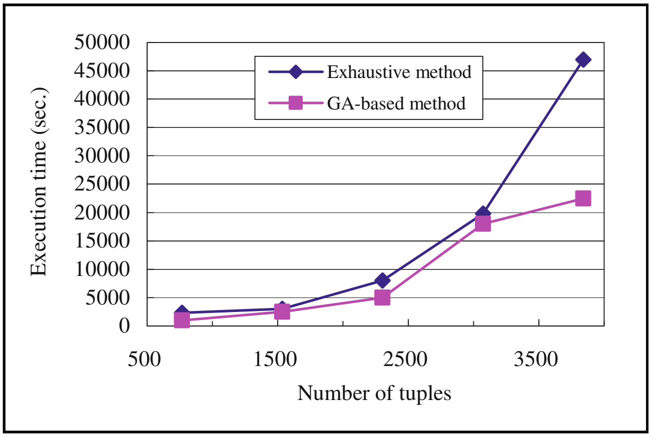


Fig. 1. Comparision of execution time between exhaustive method and GA-based method.

Not surprisingly, our GA-based method outperforms the exhaustive method. The performance gap becomes more significant when the dataset grows larger. Note that the main factor on execution time is the process of evaluating the chromosomes (data group), i.e., computing normalized CBSQS. Our method can significantly prune the number of candidate data groups, leading to fewer times of fitness evaluations.

5.2 Evaluation on Solution Correctness

In the second part of our experiments, we tested the solution correctness of our method. We refer to the study by Pearson [9], the only work known to us that provides convincing results on disguised missing data related to data group. The dataset of concern is the FDA Adverse Event Reporting System dataset [3] from January 1 to March 31 in 2004. Because the dataset suffers from high fraction of missing values in several attributes, we chose 4 attributes with the fewest missing data, including EVENT_DT, GNDR_COD, AGE, and WT and divided the attribute EVENT_DT into three attributes, say Year, Month, and Day, obtaining totally six attributes in the dataset.

According to [9], "January 1" is a common disguise value used as a surrogate for "data unknown" in entering Event Date data into the FAERS system. Similarly, the first day of other months, such as "February", "March", "April", is also very likely used as a disguise. We intended to inspect, when selecting "January 1" and "February 1" as disguise values, whether the best data groups discovered by our GA-based algorithm will consist of attributes "January" and "February", respectively. Table 3 shows the experimental results, where zero values correspond to attributes used for evaluating the degree of fitness.

Table 3. Solution comparison between GA-based approach and exhaustive method.

	v	Gender	Year	Month	Age	WT	fitness
GA-based method	*Jan. 1*	*male*	*2003*	*January*	0	0	0.8863
Exhaustive method (pruning)		*male*	*2003*	*January*	0	0	0.8863
Exhaustive method		*male*	*1998*	*January*	0	0	0.9595
GA-based method	*Feb. 1*	*male*	*0*	*February*	0	0	0.2915
Exhaustive method (pruning)		*male*	*0*	*February*	0	0	0.2915
Exhaustive method		*male*	*2000*	*February*	0	0	0.6358

In this experiment, we performed two different exhaustive approaches, with or without executing record-based pruning. The solutions generated by these two different exhaustive methods were different for "January 1" and "February 1". Specifically, the exhaustive method without record-based pruning exhibits significant better solutions than that with pruning. A further inspection showed that the projection tables $\tilde{T}_g$ corresponding to data groups found by exhaustive method without pruning for disguised values "January 1" and "February 1" only contain 18 and 20 records, respectively. Since a small $\tilde{T}_g$ loses good representation of the original dataset and leads to biased results, we chose to use the exhaustive method with pruning.

Our GA-based approach yielded the same solutions generated by the exhaustive method. For disguise value "January 1", both methods returned $g*$ composed of "January" on "Month" and "male" on "Gender", and returned "February" on "Month" and "male" on "Gender" for "February 1". However, the results do not exactly match the analysis conducted by Pearson. This is because the statistical analysis in [9] only considered a day of month, not the whole data set including other attributes, such as attribute "Gender".

6 Conclusions

The problem of detecting the data group most prone to a specific disguise value is a novel issue of detecting disguise missing data, which has not yet been addressed before. In this paper, we presented this problem and formalized it as an optimization problem by adapting the CBSQS-based heuristic [4]. We devised a GA-based approach that relies on our proposed CBSQS-derived fitness function. We also developed some effective optimization techniques to avoid unnecessary exploration of the candidate space. Experimental results showed that our method can discover the same optimal results generated by exhaustive method, and the discovered data group is analogous to previous work [9] that relied on tedious statistics based manual examinations by analyzers.

A recent work by Belen [1] has shown the benefit of replacing the CBSQS function by chi-square test based function to measure the similarity of two tables. We will investigate the effect of replacing the chi-square test based function as the fitness function.

Our developed GA-based method though can effectively discover the optimal solutions to the problem, requires lots of computations. In the future, we will pursue more efficient methods to speed up the performance.

References

1. Belen, R.: Detecting disguised missing data. Master thesis, The Middle East Technical University (2009)
2. Belen, R., Temizel, T.T.: A framework to detect disguised missing data. In: Senthil Kumar, A.V. (ed.) Knowledge Discovery Practices and Emerging Applications of Data Mining: Trends and New Domains, pp. 1–22. IGI Global, Hershey (2010)
3. FDA Adverse Event Reporting System. http://www.fda.gov/Drugs/GuidanceComplianceRegulatoryInformation/Surveillance/AdverseDrugEffects/ucm083765.htm
4. Hua, M., Pei, J.: Cleaning disguised missing data: a heuristic approach. In: Proceedings of 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 950–958 (2007)
5. Hua, M., Pei, J.: DiMaC: a system for cleaning disguised missing data. In: Proceedings of 2008 ACM SIGMOD International Conference on Management of Data, pp. 1263–1266 (2008)
6. Little, R., Rubin, D.: Statistical Analysis with Missing Data. Wiley Publishers, New York (1987)
7. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge (1996)
8. Natarajan, K., Li, J., Koronios, A.: Detecting mis-entered values in large data sets. In: Proceedings of the 4th World Congress on Engineering Asset Management, pp. 805–812 (2009)
9. Pearson, R.K.: The Problem of Disguised Missing Data. ACM SIGKDD Explor. Newslett. **8** (1), 83–92 (2006)
10. UCI Machine Learning Repository: Pima Indians Diabetes Data Set. http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes

rRNA of Alphaproteobacteria Rickettsiales and mtDNA Pattern Analyzing with Apriori & SVM

Seung Jae Lim(✉), Shin Hyo Bang, Dae Seop Kim,
and Taeseon Yoon

Hankuk Academy of Foreign Studies, Mohyeon-myeon, Cheoin-gu,
Yongin-si, Gyeonggi-do, Republic of Korea
{Tonylim0930,whoami123461,mr.kim960404,
rimehappy}@gmail.com

Abstract. The computer technology has advanced profoundly that the application seems to have no limit. Equipped with programming, our research team has created programs that could be used practically in the field of chemistry. The purpose of this paper is not on the making useful tools for science, but on using analytic tools based on computer programming to find additional evidence that supports a theory. The Suport Vector Machine (also known as its acronym, SVM) is used frequently in genetic analysis to find certain patterns in DNA sequence. This paper deals with pattern similarity between rRNA of mitochondria and that of alphaproteobacteria, which is believed to be the ancestor of the mitochondria. This theory, also known as "endosymbiotic theory" has a variety of evidences and has accepted as authentic. The pattern similarity between the two organisms' DNA sequence, which is the result of the paper would consolidate the evolutionary endosymbiosis.

Keywords: Alphaproteobacteria rickettsiales · Ribosomal RNA · Apriori · SVM · Mitochondria

1 Introduction

1.1 Endosymbiotic Theory

The advent of prokaryote cells is considered as the inception of the life on the planet, and the gathering of these prokaryote cells is presumed to form eukaryotic cell. At present, the representative prokaryotic organisms include E.coli (Escherichia coli) bacteria. The endosymbiotic theory, suggests that the aerobacteria were brought inside cells and through the long process of evolution, they settled down [1] (Fig. 1).

1.2 Supporting Evidences

1.2.1 Distinct DNA

Mitochondria has its own unique DNA, which is totally different from the cell's DNA. The shape of the DNA inside the mitochondria is circular. The prokaryote cells have

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 112–122, 2014.
DOI: 10.1007/978-3-319-13186-3_11

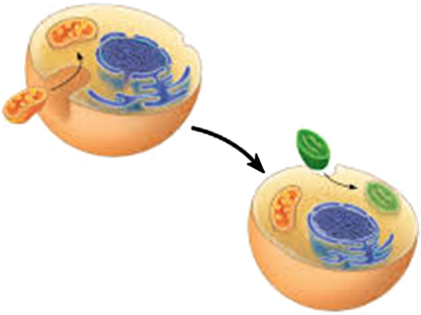


Fig. 1. Description of endosymbiotic theory [2]

circular DNA while eukaryotic cells have strongly coiled DNA. The DNA of mitochondria, so to speak, resembles that of prokaryote cells even more than the cell where the tiny organism lives in.

1.2.2 Double Membranes

Mitochondria has double membranes. This could be understood by the explanation that when aerobacteria entered the cell, the cell membrane covered the organism (Fig. 2).

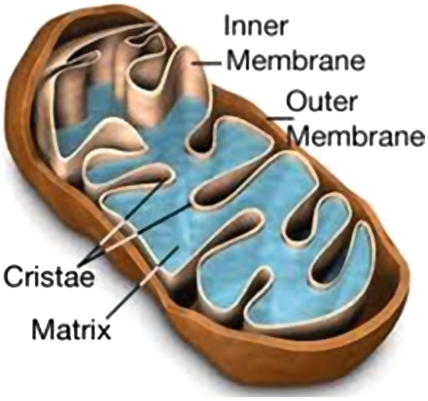


Fig. 2. Mitochondria structural features [3]

1.2.3 Genome Resemblance

As mentioned, mitochondria has its own distinct DNA. Also, according to erstwhile studies, the genomes of mitochondria have basically resemble that of the Rickettsial bacteria [4, 5].

1.3 Apriori Algorithm

Apriori algorithm is the first developed algorithm to find association between the data. Apriori algorithm approaches the data mathematically to find based on the frequency.

The basic rule used in Apriori algorithm is that all subsets of the frequency set have high frequency. The algorithm follows next four steps. First step is identifying the frequency set of data. Second step is finding the minimum support. Third is generating the candidate set. Final step is repeating second step and third step.

Apriori uses Using breadth-first search and a Hash tree structure. The algorithm counts candidate item sets efficiently. The process is as in the following. First, it generates candidate item sets of length k from item sets of length k-1. Then it prunes the candidates that contain an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. Finally, it scans the transaction database so that we can determine frequent item sets among the candidates [6]. Also Apriori uses a "bottom up" approach. In the approach, frequent subsets are extends one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm operates until no further successful extensions are found. We used 5, 7, and 9 window to find out periodicity of comparing sequences. Each of 5 window, 7 window and 9 window are data divided at every five, seven and nine amino acid. Therefore, it shows more accurate similarity by using periodicity.

1.4 Support Vector Machine (SVM)

Our research to compare DNA similarity between Rickettsia and various bacteria follows multiple steps below. First, extract DNA string bit information of various bacteria which are mostly estimated to have similar DNA information of mitochondria. Next, make algorithm to learn standard judgment, hyperplane, by using Support Vector Machine. Finally confirm similarity of optional DNA information of mitochondria based on the classifier, the learned standard judgment. Normally DNA string bit information of various bacteria from first step above is more likely to have inseparability problem which cannot separate data linearly. The approach to the inseparability problem of Support Vector Machine applies Soft Margin SVM to lower the generalization errors of the classifier. It remaps the formal non-linear data to high infinite dimensional spaces, presumably making the separation easier in the high dimensional spaces. Support Vector Machine defines this method by using a kernel function to keep the computational logics reasonable. In our paper, SVM is used to classify linear or non-linear data into predictable class, and unpredictable class. If data is spread widely so that it is uncertain to separate, SVM automatically deletes this vagueness to carry out perfect classification. Through SVM, we are able to finitely distinguish exact DNA codes of bacteria, and carry out our research [7, 8, 9].

1.5 Differences Between Apriori and Support Vector Machine (SVM)

Both Apriori and Support Vector Machine are the most representative algorithms that is used in data mining. Data mining is a process of automatically and systematically finding statistical rules or patterns in a big pile of data. However, each algorithm has its own unique characteristics as it is described above. To put it short, Apriori's role is to find a correlation between multiple item sets. In contrary, the Support Vector Machine has artificial intelligence; this relatively modern algorithm could be educated. If given certain illustrations, the algorithm learns the qualities of certain data. Subsequently, SVM operates and differentiate data based on the characteristics that it learned from the previous step [10, 11].

2 Experiment Object

2.1 Rickettsia

Table 1 explains the classification of rickettsia. Endosymbiotic theory has its evidences. Among some evidences, we've focused on genome resemblance which suggests that

Table 1. Biological classification of Rickettsia [12]

Kingdom	Bacteria
Phylum	Proteobacteria
Class	Alphaproteobacteria
Order	Rickettsiales
Family	Rickettsiaceae
Genus	Rickettsia
Species	-*Rickettsia aeschlimannii* Beati et al. 1997
	-*Rickettsia africae* Kelly et al. 1996
	-*Rickettsia akari* Huebner et al. 1946 (Approved Lists 1980)
	-*Rickettsia australis* Philip 1950 (Approved Lists 1980)
	-*Rickettsia bellii* Philip et al. 1983
	-*Rickettsia canadensis* corrig. McKiel et al. 1967 (Approved Lists 1980)
	-*Rickettsia conorii* Brumpt 1932 (Approved Lists 1980)
	-*Rickettsia felis* Bouyer et al. 2001
	-*Rickettsia heilongjiangensis* Fournier et al. 2006
	-*Rickettsia helvetica* Beati et al. 1993
	-*Rickettsia honei* Stenos et al. 1998
	-*Rickettsia japonica* Uchida et al. 1992
	-*Rickettsia massiliae* Beati and Raoult 1993
	-*Rickettsia montanensis* corrig. (ex Lackman et al. 1965) Weiss and Moulder, 1984
	-*Rickettsia parkeri* Lackman et al. 1965 (Approved Lists 1980)
	-*Rickettsia peacockii* Niebylski et al. 1997
	-*Rickettsia prowazekii* da Rocha-Lima 1916 (Approved Lists 1980)
	-*Rickettsia rhipicephali* (ex Burgdorfer et al. 1978) Weiss and Moulder, 1988
	-*Rickettsia rickettsii* (Wolbach 1919) Brumpt 1922 (Approved Lists 1980)
	-*Rickettsia sibirica* Zdrodovskii 1948 (Approved Lists 1980)
	-*Rickettsia slovaca* Sekeyová et al. 1998
	-*Rickettsia tamurae* Fournier et al. 2006
	-*Rickettsia typhi* (Wolbach and Todd 1920) Philip 1943 (Approved Lists 1980)

mitochondria and bacteria have similar genomes. Formal studies of comparing basic sequences (DNA) between mitochondria and rickettsia revealed that there exists great resemblance between them. Specifically, the genome sequence of *Rickettsia prowazekii* and that of mitochondria presented similarities [13]. Therefore, we assumed that other species included in rickettsia would have something in common with mitochondria. Also, recognizing patterns of rickettsia and comparing results with mitochondrial DNA (mtDNA) would reveal whether resemblance among them exists or not. Plus, we've focused on ribosomal RNA which takes charge of organisms' protein (DNA, mRNA, tRNA, rRNA etc.) production. Furthermore, one of the evidence of endosymbiotic theory is that mitochondrial ribosome and bacterial ribosome have resemblances with patterns and basic sequences. Thus, we've aspired to recognize 16s ribosomal RNA (rRNA) pattern of *rickettsia* and mitochondria employing SVM (Support Vector Machine). Among many species of *rickettsia*, four of them whose 16s rRNA sequences are available were downloaded from the Genbank (The National Center for Biotechnology Information, http://www.ncbi.nlm.nih.gov) [14]. They are *rickettsia canadensis, rickettsia prowazekii, rickettsia rickettsii*, *rickettsia typhi.*

2.2 Mitochondrial DNA

Mitochondrial DNA (mtDNA or mDNA) is the DNA of mitochondria. Normally, mitochondria convert chemical energy from food into adenosine triphosphate (ATP). Mitochondria have their own DNA and the chloroplast as well. Most of species, including humans, mtDNA is inherited from the mother. Also, mtDNA has been

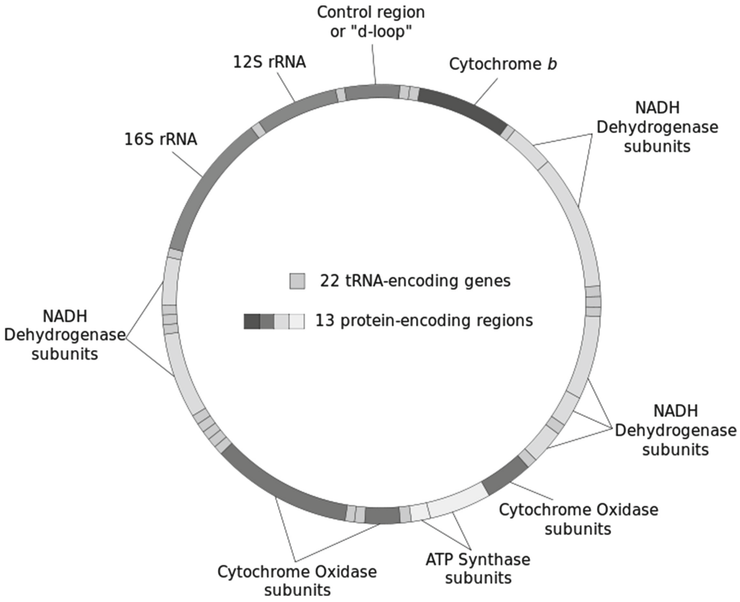


Fig. 3. Mitochondrial DNA [15]

influenced from a lot of organisms and individual. According to Fig. 4, 16S rRNA and 12S rRNA exists. rRNA, which is ribosomal ribonucleic acid, is inevitable for protein synthesis in organisms (Fig. 3).

3 Experiments

3.1 Comparing RRNA of Mitochondria and Rickettsia

As mentioned, rRNA of selected samples (*Rickettsia canadensis, Rickettsia prowazekii, Rickettsia rickettsii*, *Rickettsia typhi*) was compared with that of Homo sapiens (Homo sapiens isolate S1 mitochondrion rRNA (12S&16S)) employing apriori. The first experiment was conducted using Apriori algorithm. The rRNA sequence was input in the algorithm. Then, Apriori counts the number of certain amino acid which includes amino acid G from M. The number of certain amino acid could be interpreted as a pattern, thus could be an appropriate measure that could be used to compare its relevance with another rRNA sequence. The experiments will be progressed under 5-window, 7-window and 9-window.

3.2 Classification of Mitochondria and Rickettsia Using Multiclass SVM

We will use multi-class support vector machine (SVMmulticlass) which uses the multi-class formulation All of experiments will employ 10-fold cross-validation to gain higher accuracy. The algorithm learned unique qualities of certain rRNA at the first place. Then, the supposedly similar rRNA sequence is input in the algorithm and the SVM starts differentiating the data based on the learned qualities. The harder the separating process, the more similar the two rRNA are. Data will be divided into 10 different sets, and when 9 data sets are used as a training data which means teaching proccess, 1 data set is considered as test data sets. Thus, 10 data sets are all used as test data for one time respectively, which means that 10 experiments were made.

4 Conclusion

4.1 Results

4.1.1 Apriori

According to Fig. 4, rickettsia (*Rickettsia canadensis, Rickettsia prowazekii, Rickettsia rickettsii*, *Rickettsia typhi*) show consistency in 12S ribosomal RNA. Rickettsia rickettsii appeared to be different among some amino acids, but overall inclination is about the same in 5-window, 7-window and 9-window. Also, homo sapiens isolate S1 mitochondrion 12S rRNA shown more resemblance when it comes to bigger window. 7-window and 9-window shown almost same tendency of amino acid. Sapiens mitochondrion 12S rRNA shown high common ground when amino acids are A, S, T, D, I, Q, N, F, Y and M. The rest amino acid revealed differences. However, majority of amino acids shown similar patterns (Fig. 5).

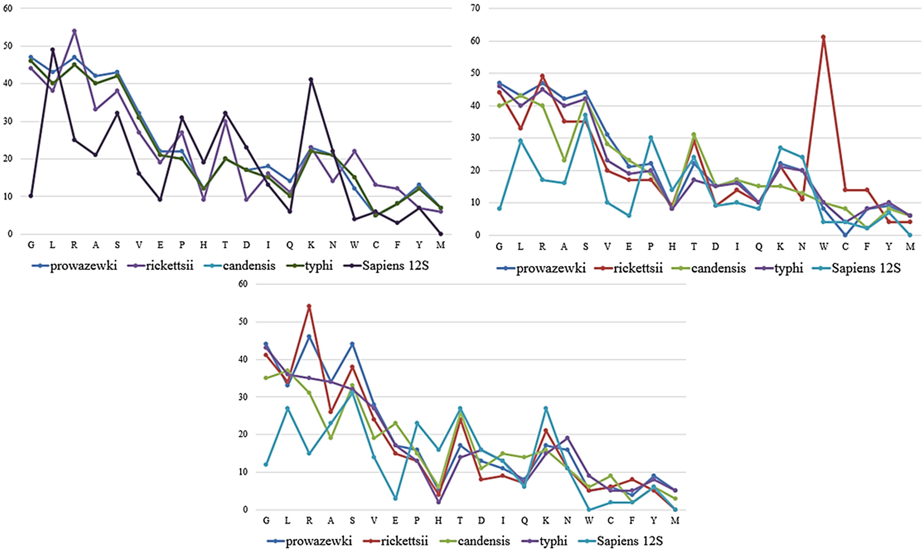


Fig. 4. Rickettsia & Sapiens 12S rRNA comparison, 5, 7, 9-window (numerically ordered)

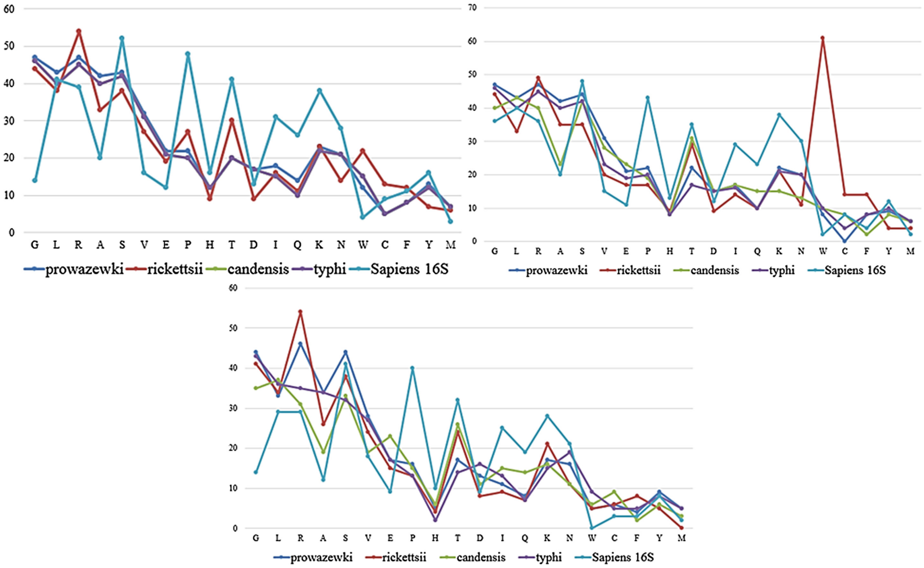


Fig. 5. Rickettsia & Sapiens 16S rRNA comparison, 5, 7, 9-window (numerically ordered)

Compared with the result of rickettsia and sapiens 12S rRNA comparison, that of 16S rRNA showed less similarities. According to Figs. 7, 8 and 9, especially, among amino acids such as G, A, P, I and K revealed huge differences in all window. However, majority of amino acids shown similar tendency. Synthetically, considering

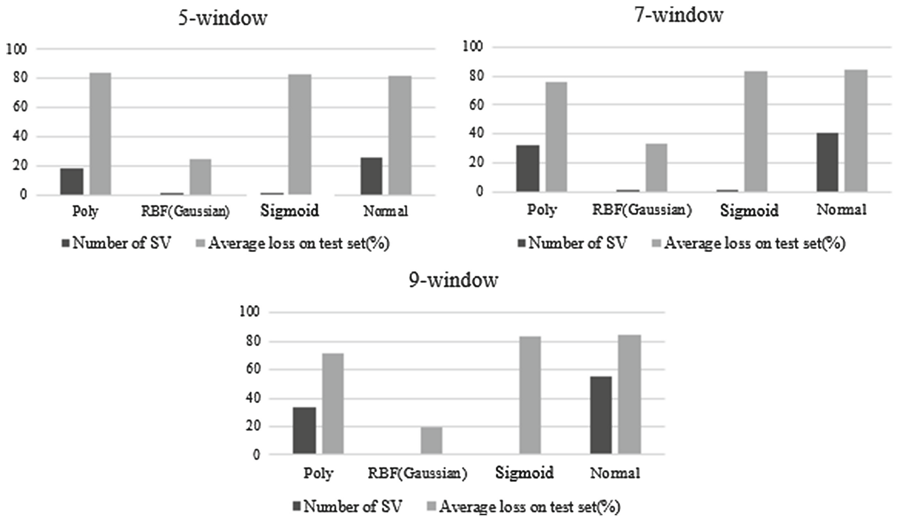


Fig. 6. Classification result using multiclass SVM

all of the result using apriori algorithm, there exists resemblance between bacteria rRNA and that of mitochondria, Also, according to result, typhi and canadensis appeared to shown the most similar patterns of mitochondria.

4.1.2 Support Vector Machine

Figure 6 is the result of SVM on each 5-window, 7-window and 9-window. It's noticeable that three functions' (Polynomial, Sigmoid, Normal) average loss on test set turned out to be high, approximately 80 % on average, except RBF (Gaussian) marked low, approximately 20 %. It raises doubts why RBF shows different tendency. There can be 2 reasons. First, it's the indicator of the fact that sequence wasn't classified well. From this it could be said that similarity between rRNA of mitochondria and that of rickettsia is high. Secondly, due to RBF's feature of discontinuity, it has tendency of revealing adversity on classifying some specific classes and non-classified data would be abandoned. Considering these facts, RBF has tendency of showing high rates when it comes to similar sequences. In conclusion, classification result using multiclass SVM supports endosymbiotic theory showing that rRNA of mitochondria and bacteria have noticeable similarities. The result of multiclass SVM shows that mitochondria and rickettsia have noticeable similarities. However, since the multiclass SVM has its original feature to have inseparable results, we've decided to make additive one-on-one experiments with SVM (Figs. 10, 11, 12).

The apriori experiment suggested that the Typhi and Candadensis showed noticeably similar patterns with mitochondria. According to the result of SVM experiment, Rickettsia typhi and Rickettsia canadensis rRNA showed the best accordance with mitochondria rRNA. We assumed that typhi, canadensis and mitochondria are whole different organisms. We've done experiments of classifying typhi and Canadensis rRNA with mitochondria 12S rRNA and 16S rRNA using SVM with various functions. Test and training were same as formal multiclass SVM experiment.

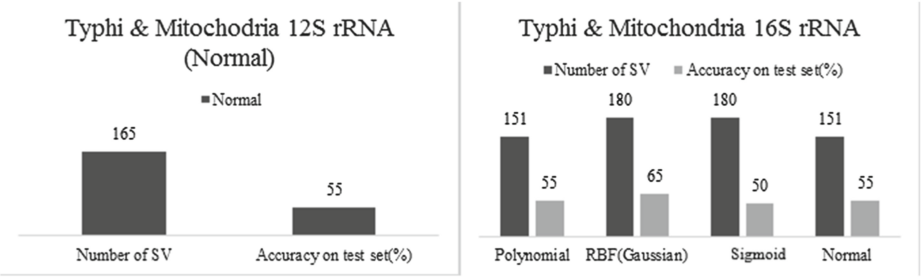


Fig. 7. Typhi & Mitochondria rRNA classification using SVM, 5-window

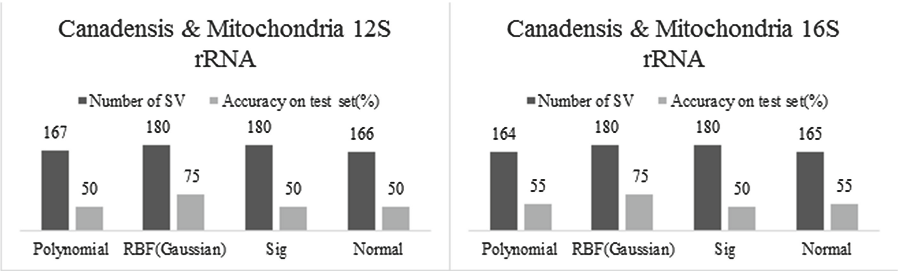


Fig. 8. Canadensis & Mitochondria rRNA classification using SVM, 5-window

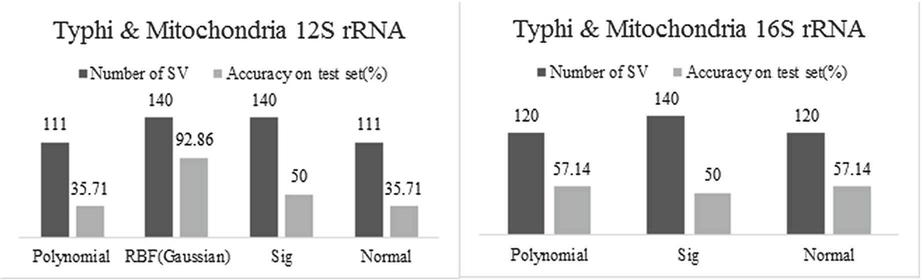


Fig. 9. Typhi & Mitochondria rRNA classification using SVM, 7-window

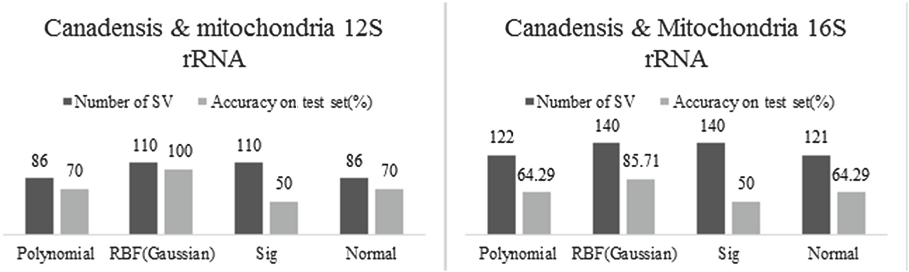


Fig. 10. Canadensis & Mitochondria rRNA classification using SVM, 7-window

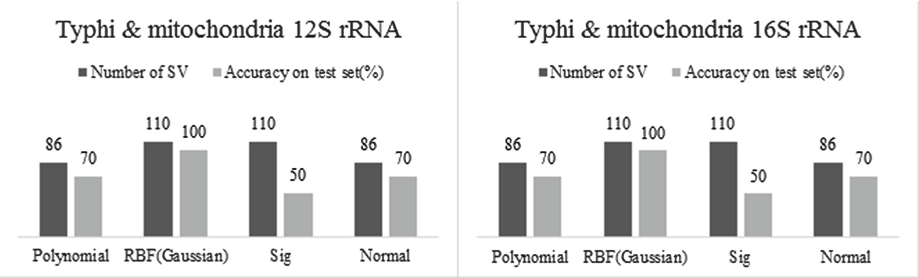


Fig. 11. Typhi & Mitochondria rRNA classification using SVM, 9-window

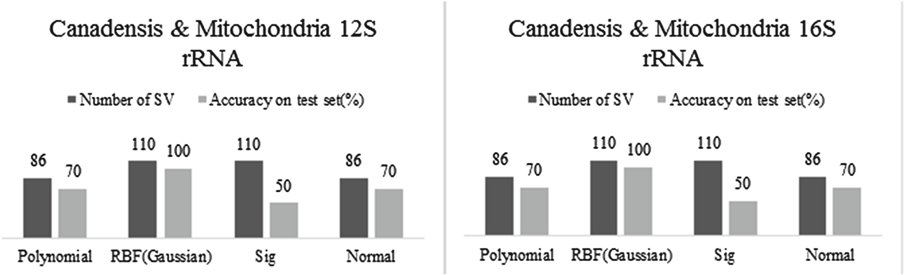


Fig. 12. Canadensis & Mitochondria rRNA classification using SVM, 9-window

Unlike the hypothesis we made, the two bacteria showed a notable resemblance with mitochondrial DNA, according to a set of experiments. In a few RBF tests, it even showed a 100 % match. Other functions also showed an approximate resemblance as the amount of information increases (the window number). In conclusion, it's valid to say that the protein structure of mitochondria has a lot in common with bacteria, which is the key evidence that supports endosymbiotic theory. Plus, it can also designate that there exist resemblance between bacteria and mitochondria, while there exist differences among them. To be specific, according to the result of apriori and support vector machine, bacteria and mitochondria showed significant resemblance of amino acidic patterns, however, there are unique features of each bacteria which could be interpreted as they evolved different amino acidic features after endosymbiosis.

4.2 Expectations

We had studied deeply into similarity between rRNA of mitochondria, and that of alphaproteobacteria(which is strongly assumed to be the aerobic bacteria). Using bioinformatical methods (apriori algorithm, and support vector machine) we found supporting evidences for endosymbiotic theory. The result shows that rRNA of mitochondria, and alphaproteobacteria have comparatively high similarity. This fact strongly supports endosymbiotic theory as another indisputable evidence explaining biological characteristic of mitochondria in the domain of genetics. This paper is valuable for its role as finding genetic evidence of endosymbiotic theory with informatics. It is expected that we can achieve similar bioinformatic results from the chloroplast which is widely known as another representative example of endosymbiotic theory.

References

1. Endosymbiotic Theory. http://wikipedia.org/
2. Reece, J.B., et al.: Campbell Biology, Chapter 6, 9th edn, p. 109 (2013)
3. Mitochondria Structural Features. http://www.ccwcs.org/
4. Bay, K., et al.: Mitochondria share an ancestor with SAR11, a globally significant marine microbe. Science Daily, 25 July 2011. Accessed 26 July 2011
5. Thrash, J.C., et al.: Phylogenomic evidence for a common ancestor of mitochondria and the SAR11 clade. Scientific reports (2011)
6. Reddy, E.M.: Effective classification using parallel apriori: pattern analysis for effective classification using parallel apriori (2012)
7. Yamashita, H., Tanaka, S.: An Introduction to Support Vector Machines, pp. 6–8 (2001)
8. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines, 6th edn., pp. 34–66 (2008)
9. Onoda, T.: Support Vector Machine (Science of Intelligence), 2nd edn., pp. 11–108 (2008)
10. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, Santiago, Chile, pp. 487–499, September 1994
11. Bayardo Jr., R.J.: Efficiently mining long patterns from databases. In: ACM SIGMOD Record, vol. 27(2). ACM (1998)
12. Encyclopedia Of Life. http://eol.org/pages/3349/overview
13. Andersson, S.G., Zomorodipour, A., Andersson, J.O., Sicheritz-Pontén, T., Alsmark, U.C., Podowski, R.M., Näslund, A.K., Eriksson, A.S., Winkler, H.H., Kurland, C.G.: The genome sequence of Rickettsia prowazekii and the origin of mitochondria. Nature **396**(6707), 133–140 (1998)
14. The National Center for Biotechnology Information. http://www.ncbi.nlm.nih.gov/
15. Sykes, B.: Mitochondrial DNA and human history. The Hu man Genome. Wellcome Trust. Accessed 5 February 2012

Mobile Data Management, Mining, and Computing on Social Networks

Team Formation with the Communication Load Constraint in Social Networks

Yui-Chieh Teng, Jun-Zhe Wang, and Jiun-Long Huang(✉)

Department of Computer Science, National Chiao Tung University,
Hsinchu City, Taiwan, ROC
yuichey@gmail.com, {jzwang,jlhuang}@cs.nctu.edu.tw

Abstract. Given a project requiring a set of skills, the team formation problem in social networks aims to find a team that can cover all the required skills and has the minimal communication cost. Previous studies considered the team formation problem with a leader and proposed efficient algorithms to address the problem. However, for large projects, a single leader is not capable of managing a team with a large number of team members. Thus, a number of leaders would be formed and organized into a hierarchy where each leader is responsible for only a limited number of team members. In this paper, we propose the team formation problem with the communication load constraint in social networks. The communication load constraint limits the number of team members a leader communicates with. To solve the problem, we design a two-phase framework. Based on the proposed framework, we first propose algorithm Opt to find an optimal team, under the communication load constraint, with minimal communication cost. For large social networks, we also propose algorithm Approx to find a nearly-optimal team. Experimental results show that algorithm Opt is able to find optimal teams and is more efficient than the brute-force algorithm. In addition, when nearly-optimal teams are acceptable, algorithm Approx is much more scalable than algorithm Opt for large social networks.

Keywords: Team formation · Degree-constrained minimum spanning tree · Social network

1 Introduction

To carry out a project with various required skills, it is usually necessary to form a team of experts that are able to cover the skill set. The fulfilment of the required skills for a project is fundamental to accomplish the project. However, a more important key to the success of the project is whether the experts in the team can effectively communicate and collaborate with each other. Thus, given a project, it is desirable to organize a team of experts possessing all the skills required for the project and with minimal communication cost.

Lappas et al. [9] were the first to consider the communication factor of team formation in social networks. They presented two communication cost metrics,

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 125–136, 2014.
DOI: 10.1007/978-3-319-13186-3_12

namely *diameter communication cost* and *minimum spanning tree cost*, to evaluate communication effectiveness of a team. Kargar and An [8] proposed to use the *sum of distances*, a more stable cost metrics, to define the communication cost of a team. They also considered the team formation problem with a leader where the leader is responsible for coordinating all team members and each team member directly communicates with the leader. To measure the communication cost of a team with a leader, they introduced *leader distance* which is the sum of the shortest distances between the leader and the corresponding team member for each required skill. A brute-force algorithm was developed to identify the best leader and the corresponding team. Juang et al. [7] proposed two algorithms, called algorithm BCPruning and algorithm SSPruning, to accelerate the discovery of the best leader and team. These two algorithms were shown to be more efficient and scalable than the algorithm proposed in [8].

Although employing a good leader managing and coordinating team members is beneficial, a single leader is not sufficiently capable of administering a large project requiring a number of experts since the leader may not have enough time to communicate with all team members. For a large project in the real world, the experts usually would be divided into different groups based on the tasks of the project and the skills of the experts. As such, instead of a single leader, a number of leaders are employed to co-work for a large project where (1) these leaders would be organized into a hierarchical structure (i.e., a tree with height larger than two) and (2) any leader communicates with no more than d leaders at the next low level or team members. d is referred to as the *communication load constraint*. In other words, finding the team and hierarchy with minimal communication cost under the communication load constraint is crucial for a large project in practice. Such characteristic distinguishes our paper from others.

In view of this, we propose in this paper the team formation problem with the communication load constraint in social networks. Specifically, the proposed problem is to discover the team and hierarchy that covers the required skills, has the minimal communication cost and guarantees each leader to have the predefined, acceptable load. To facilitate the process of identifying the desired team and hierarchy, we present a two-phase framework consisting of the team generation phase and the hierarchy establishment phase. The team generation phase is to find all the teams *qualified* for a given project (i.e., the team members possess all the required skills). In the hierarchy establishment, the hierarchy with the minimal communication cost of each team is created and the team and hierarchy having the minimal communication cost is then determined. Note that the hierarchy of a team meeting the load constraint and incurring the minimal communication cost is recognized as the *degree-constrained minimum spanning tree*, a well-known NP-complete problem. To solve the team formation problem with the communication load constraint, we first develop a naive algorithm, called algorithm Brute-Force, to find the best team and hierarchy by enumerating all the eligible teams and hierarchies. By analysing the breakdown of the execution time of algorithm Brute-Force, we observe that the hierarchy establishment phase accounts for the execution time since finding the minimal spanning trees with the load constraint of all the qualified teams is very time-consuming. Thus, we design algorithm Opt to employ the lower bounds of the communication cost to prune some qualified teams in the first phase, thereby reducing the number of teams evaluated in the second phase. Although algorithm Opt is more efficient than algorithm Brute-Force, algorithm Opt suffers from heavy computation in large social networks. Since nearly-optimal solutions have been widely acceptable in many situations, we devise algorithm Approx to find nearly-optimal teams by applying the 2-opt change approximation [10] to construct nearly-minimum spanning trees with degree constraint. Experimental results show that algorithm Opt outperforms algorithm Brute-Force in terms of the execution time. Moreover, algorithm Approx is much more scalable than algorithm Opt at a cost of small communication cost increase (less than 10 % in our experiments).

The remainder of this paper is organized as follows. Section 2 presents a survey of related work in the literature of team formation problem, followed by the formulation of the team formation problem under the communication load constraint. We then describe algorithm Brute-Force, algorithm Opt and algorithm Approx in Sect. 3 in detail. The experimental results are reported in Sect. 4. Finally, Sect. 5 concludes this paper.

2 Preliminaries

2.1 Related Work

Team Formation Problem in Operation Theory. The formation of the multi-functional teams, which are the teams required a set of skills to be handled, is crucial and has gained attention in recent years. Zakarian and Kusiak combined the different factors of the team selection problem into a hierarchical structure [11]. They also provided the quality functional development method and used the integer linear program (ILP) to model the problem. Fitzpatrick and Askin developed and tested mathematical models for formation of effective human teams based on the Kolbe Conative Index and presented a programming formulation for the problem [4]. Chen and Lin proposed a working relation model with respect to the multi-functional knowledge and the capability of each individual, and the co-work relation between members [3]. Different from the work in [3], Baykasoglu et al. used the fuzzy optimization model and the problem is solved by the annealing algorithm [2]. Gaston et al. considered about not only the impact of the network structures on the team performance but the network structure between individual [6].

Team Formation Problem in Social Networks. Lappas et al. [9] first addressed the problem with considering the individuals of social networks. They defined each node in a social network to represent an expert having a set of skills and the communication cost between two experts is the weight of the edge connecting them in the social network. If no such edge exists, the distance of the shortest path between these two experts is used to measure the communication cost between them. The communication cost indicates how effectively the two experts collaborate. Thus, the smaller the communication cost is, the easier they can co-work. They proposed to use the *minimal spanning tree* and the *diameter* to measure the communication cost of a team. They also proved the NP-hardness of the two communication cost metrics for the team formation problem and proposed the approximation algorithms. In [8] Kargar and An argued that the communication cost metrics proposed in [9] were insensitive thus introduced a new communication cost metric: *sum of distance* to reveal all the skills needed to be communicated between any two experts. Furthermore, they also introduced the team formation problem with a leader. The leader is responsible for the team and is always coordinating with other team members. Kargar and An also proposed in [8] the *leader distance* to measure the communication cost of a team with a leader. In [7] Juang et al. proposed two efficient algorithms, algorithm BCPruning and algorithm SSPruning, for the team formation problem with a leader and these two algorithms outperform the previous work proposed in [8]. However, the team organizations in previous works, including [7,8], are trees with height two, and the communication cost of a team lies on the communication responsibility of the leader. In practices, a leader can only directly communicated with limited team members, and thus, organizing a team into a hierarchy (a tree being able to with height larger than two) satisfying the communication cost constraint is necessary for large projects.

Team Formation Problem with Load Balance. In light of the fair work load for each member in the team, Anagnostopoulos et al. [1] considered the multi-project situation in team formation problem. The scenario in the work is to dispatch projects to people that can deal with more than one project at same time. But the load balance is considered and every expert has the upper bound of the number of projects involved. In addition to the fairest workload, they also find the team that has the minimal communication cost and thus making the problem more applicable in the real world. Nevertheless, the problem only probed into the multi-project allocation in on-line team formation and focused on minimizing the communication cost while having a fair work for every expert. In a real world, the communication between experts is as important as the technical expertise of each individual and thus the communication load of experts should not be neglected.

2.2 Problem Formulation

Given a group of experts $X = \{x_1, x_2, \ldots, x_n\}$, the skill set composed of the skills which all the experts in X have is denoted by $S = \{s_1, s_2, \ldots, s_m\}$. Let $skill(x)$ be the set of the skills that expert x possesses. We model an expert social network as an undirected graph $G(X, E)$ where each edge in E, say (x_i, x_j), indicates that x_i and x_j had co-worked with each other before. In addition, the weight of (x_i, x_j) is the communication cost of x_i and x_j.

Definition 1. *Given a project $P \subseteq S$, a team $T \subseteq X$ is qualified for P if for each skill, say s in P, there exists one expert in T, say x, so that $s \in skill(x)$.*

Given a team T and an expert social network G, the communication graph is defined as below.

Definition 2. *The communication graph of* T *on* G*, denoted as* $G_T(T, E_T)$*, is a weighted and undirect complete graph, where the weight of each edge, say* (x_i*,* x_j)*, in* E_T *indicates the communication cost between experts* x_i *and* x_j*.*

If x_i and x_j had co-worked in the past, the edge (x_i, x_j) should exist in E and the weight of the edge (x_i, x_j) in G_T is set to the weight of the edge (x_i, x_j) in G. Otherwise, when the edge (x_i, x_j) is not in E, similar as the prior work [7,8], the weight of the edge (x_i, x_j) in G_T is defined as the distance of the shortest path between x_i and x_j in G.

Definition 3. *Given a communication graph* $G_T(T, E_T)$*, the hierarchy of* T *under the communication load constraint* d *is a spanning tree where the degree of each node is at most* d*. In addition, the communication cost of the hierarchy is defined as the sum of the weights of the edges in the hierarchy.*

Definition 4. *Given an expert social graph* $G(X, E)$ *and a project* P*, the team formation problem with the communication load constraint* d *is to find a team* T *and the corresponding hierarchy with the minimal communication cost under the communication cost constraint.*

3 Proposed Algorithms

3.1 Algorithm Brute-Force

To solve the team formation problem with the communication load constraint, we propose a two-phase framework that comprises the team generation and hierarchy establishment phases below.

1. *Team generation phase:* Find all teams qualified for the given project.
2. *Hierarchy establishment phase:* Determine the team and hierarchy with the minimal communication cost under the communication load constraint by establishing the hierarchy with minimal communication cost for each team discovered in the team generation phase.

Based on the proposed two-phased framework, we develop a naive algorithm, called algorithm Brute-Force, to solve the team formation problem with the communication load constraint. Algorithm Brute-Force first use the method used in [7] to generate all the qualified teams in the team generation phase. Then, for each qualified team, algorithm Brute-Force generates the degree-constrained minimum spanning tree by enumeration. Note that the hierarchy establishment is the well-known NP-complete problem, the degree constrained minimum spanning tree problem on $G_T(T, E_T)$. Finally, algorithm Brute-Force takes the team and the spanning tree with minimum communication cost (hierarchy) as the answer to the team formation problem.

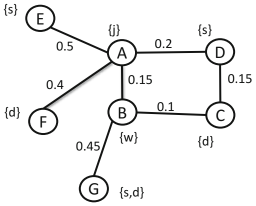


Fig. 1. A social network of experts

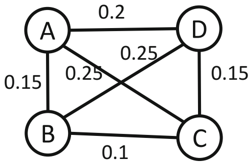


Fig. 2. The communication graph of team (A, B, C, D)

We use Fig. 1 and Table 1 as an example to illustrate the process of algorithm Brute-Force. Figure 1 shows an expert social network where a circle represents an expert and the letter(s) next to a circle indicate(s) the skills the expert possesses. The letter s, w, d and j mean the skills "software engineering," "web programming," "data mining" and "java," respectively. The weight of an edge represents the communication cost between the two experts connected by the edge. Consider a project $P = \{j, s, d, w\}$ that requires four skills, namely java, software engineering, data mining and web programming. The communication load constraint is set to 2.

Table 1 shows the process of algorithm Brute-Force. In the team generation phase, five qualified teams ((A, B, C, D), (A, B, C, E), (A, B, F, D), (A, B, F, E) and (A, B, G)) are found. In team (A, B, C, D), A, B, C and D are responsible for java, software engineering, data mining and web programming, respectively. To calculate the minimum cost of team (A, B, C, D), algorithm Brute-Force first generates the communication graph of team (A, B, C, D), as shown in Fig. 2. Then, all hierarchies (i.e., degree-constrained spanning trees), under the communication load constraint, of the team are enumerated. After calculating the communication cost of each generated hierarchy, we can find the degree-constrained minimum spanning tree of the minimal communication cost $0.15 + 0.15 + 0.1 = 0.4$. The communication costs of other qualified teams are also calculated in a similar manner. Finally, algorithm Brute-Force reports that the best team is (A, B, C, D) of communication cost 0.4.

3.2 Algorithm Opt

Since enumerating all the hierarchies of each qualified team, algorithm Brute-Force obviously performs poorly in large social networks. In order to improve the

Table 1. The process of algorithm Brute-Force

Skills				Minimum
Java	Software engineering	Data mining	Web programming	communication cost
A	B	C	D	0.4
A	B	C	E	0.75
A	B	F	D	0.75
A	B	F	E	1.05
A	B	G	G	0.6

Table 2. Breakdown of execution time of algorithm Brute-Force (unit (ms))

	Number of required skills				
	3	4	5	6	7
Team generation	41594	57045	87750	102258	122345
Hierarchy establishment	5209834	7175670	10758594	12701720	3160379

performance, we conduct a preliminary experiment to observe the distributions of the execution time of algorithm Brute-Force. The social network consists of 30 nodes and 58 edges. The communication load constraint is set to 2 while the number of required skills ranges from 3 to 7. The experimental result is shown in Table 2. We can see that hierarchy establishment phase accounts for more than 99 % of the total execution time. As the number of nodes in the social network increases, more teams will be generated in the team generation phase. More importantly, the execution time of the hierarchy establishment phase will increase significantly since the problem (i.e., degree-constrained minimum spanning tree problem) addressed in the hierarchy establishment is NP-complete [5]. Such an observation motivates us to proposed algorithm Opt to reduce the number of teams evaluated in the hierarchy establishment phase. Specifically, algorithm Opt takes advantage of the currently best communication cost to prune those qualified teams that cannot be of the minimal communication cost, thereby substantially saving the execution time of hierarchy establishment.

To avoid the high cost of finding the minimal-cost hierarchy (a degree-constrained minimum spanning tree) of a qualified team, we exploit the property of spanning trees that each node, except the root, contributes one edge to the spanning tree. In other words, there are $k-1$ edges in a spanning tree formed by a team consisting of k experts. With the property, we introduce the following definitions.

Definition 5. *Given a team T and the corresponding communication graph $G_T(T, E_T)$, the lower bound of the communication cost contributed by an expert x_i, denoted as $LB(x_i)$, is defined as*

$$LB(x_i) = \min_{\forall (x_i, x_j) \in E_T} \textit{weight of } (x_i, x_j).$$

Table 3. The process of algorithm Opt

Skills				LB(T)	Minimum communication cost
Java	Software engineering	Data mining	Web programming		
A	B	C	D		0.4
A	B	C	E	0.35	0.75
A	B	F	D	0.5	-
A	B	F	E	0.5	-
A	B	G		0.3	0.6

Based on Definition 5, we can derive the low bound of a team T on $G_T(T, E_T)$ below.

Definition 6. *Given a team T and the corresponding communication graph $G_T(T, E_T)$, the low bound of the communication cost of T, denoted as $LB(T)$, can be formulated as*

$$LB(T) = \sum_{\forall x_i \in T} LB(x_i) - \max_{\forall x_i \in T} LB(x_i).$$

We use Fig. 2 and Table 3 to illustrate the process of algorithm Opt. Similar to algorithm Brute-Force, algorithm Opt first generates all qualified teams and the qualified teams are listed in Table 3. In the hierarchy establishment phase, algorithm Opt first calculates the communication cost of team (A, B, C, D) by the method used in algorithm Brute-Force and obtains the communication cost of (A, B, C, D) to be 0.4. Algorithm Opt marks (A, B, C, D) as the candidate of the best team. Before proceeding to calculate the communication cost of team (A, B, C, E), algorithm Opt calculates the lower bound of the communication cost of team (A, B, C, E). Since the lower bound of the communication cost of team (A, B, C, E) is less than the communication cost of the candidate (A, B, C, D), algorithm Opt finds the hierarchy of the minimum communication and calculates the actual minimum communication cost of (A, B, C, E). Algorithm Opt continues to evaluate (A, B, C, E). Since the low bound of the communication cost of (A, B, C, E) is larger than the communication cost of the candidate best team (A, B, C, D), (A, B, C, E) can be pruned without identifying the hierarchies. Algorithm Opt processes the remaining teams following the above process and finally determines the best team to be (A, B, C, D).

3.3 Algorithm Approx

As mentioned earlier, the hierarchy establishment problem is recognized as the degree-constrained spanning tree problem, a well-known NP-complete problem [5]. Thus, it is computationally expensive to discover the hierarchy of the minimum communication cost for a team. Although algorithm Opt is able to reduce the time of the hierarchy discovery by pruning those qualified teams that cannot be of minimum communication cost, algorithm Opt still suffers from heavy computation on

Table 4. The conferences selected from the DBLP dataset

Category	Conferences
Database	SIGMOD, VLDB, ICDE, ICDT, EDBT, PODS
Data mining	KDD, WWW, SDM, PKDD, ICDM
Artificial intelligence	ICML, ECML, COLT, UAI
Theory	SODA, FOCS, STOC, STACS

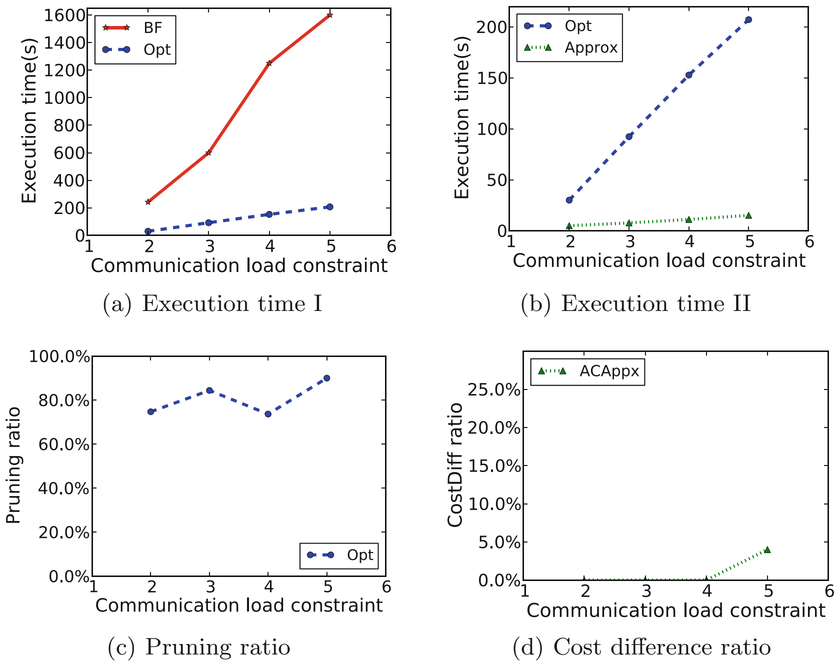
hierarchy establishment for the other teams. In a real world, approximate answers are acceptable for those NP-complete problems in many situations. As such, we devise algorithm Approx to obtain nearly-optimal teams and hierarchies. To find nearly-optimal hierarchies of qualified teams, we adopt the 2-opt algorithm proposed in [10] to find a nearly-optimal the degree-constrained minimum spanning tree for each qualified team. To be specific, the main difference between algorithm Approx and algorithm Opt is that algorithm Approx utilizes an approximate algorithm, 2-opt algorithm, for hierarchy establishment of each qualified team rather than the enumeration method used in algorithm Opt to achieve faster problem resolving.

4 Performance Evaluation

4.1 Experimental Setup

To reveal the real social network information, we extract the expert social network from the DBLP bibliography on November 7, 2011. Referring to [7,8], the DBLP dataset is composed of only the papers published in the prestigious conferences in the areas of database, data mining, artificial intelligent and theory. The selected conferences of the four areas are listed in Fig. 4. To generate the expert social network, we select only the authors publishing at least three papers. Two experts (authors) are connected together if they have worked together with at least two papers. Let p_{x_i} be the set of papers published by expert x_i. Same as [7], for each pair of two connected experts x_i and x_j, the communication cost between x_i and x_j is defined as $1 - \frac{p_{x_i} \cap p_{x_j}}{p_{x_i} \cup p_{x_j}}$.

The proposed three algorithms are implemented in Java and are executed on a PC with an Intel i7 2.93 GHz CPU and 4GB memory. For each experiment, 30 projects are randomly generated by the same method used in [7]. Because the problem addressed in the hierarchy establishment phase is NP-complete, algorithm Brute-Force and algorithm Opt are expected to of much higher execution time. To speed up algorithm Brute-Force and algorithm Opt, we pre-build all degree-constrained spanning trees for small scenarios (i.e., skill number smaller than or equal to 7 and degree constraint smaller than or equal to 6). Besides, the communication cost between any two experts is pre-computed as the prior works [7,8]. We employ the following three performance metrics to evaluate the performance of the proposed algorithms.



(a) Execution time I

(b) Execution time II

(c) Pruning ratio

(d) Cost difference ratio

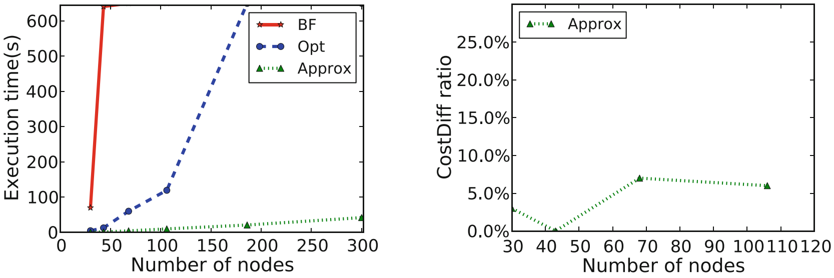
Fig. 3. Impact of the communication load constraint

- Query execution time: The query execution time is defined as the CPU time of each algorithm.
- Pruning ratio: Pruning ratio is used to measure the performance of the pruning strategy used in algorithm Opt. Let n and n_p be the number of teams which can handle the project and the number of teams that are pruned without sent to the hierarchy establishment phase. The pruning ratio is defined as $\frac{n_p}{n}$.
- Cost difference ratio: The cost difference ratio is used to measure the quality of the solutions found by algorithm Approx, and is defined as $\frac{c_{Approx}-c_{Opt}}{c_{Opt}}$ where c_{Opt} and c_{Approx} are the communication costs of the teams found by algorithm Opt and algorithm Approx, respectively.

We use 'BF', 'Opt' and 'Approx' to represent algorithm Brute-Force, algorithm Opt and algorithm Approx, respectively, in the following subsections.

4.2 Impact of the Communication Load Constraint

In this experiment, we measure the impact of the communication load constraints on the proposed algorithms by varying the constraint from 2 to 5. The number of the required skills for a project is set to 6. The experimental result is shown in Fig. 3. As observed in Fig. 3(a), algorithm Opt is more efficient than algorithm Brute-Force due to the effect of pruning. In addition, the speedup of



(a) Execution time

(b) Cost difference ratio

Fig. 4. Impact of the size of the social networks of experts

algorithm Opt over algorithm Brute-Force increases as the increase of the communication load constraint. As shown in Fig. 3(c), it is interesting that the pruning ratio is around 70 %–90 % as the communication load constraint increases from 2 to 5. The reason is that although the pruning ratio seems not of explicit relationship to the communication load constraint, the execution time of hierarchy establishment phase increases with the increase of the communication load constraint. On the other hand, with an approximation algorithm in the hierarchy establishment phase, it is obvious that algorithm Approx is much more scalable than algorithm Opt. The result depicted in Fig. 3(b) agrees with the intuition. Figure 3(d) reveals that the cost difference ratio is smaller than 6 %, showing that the trade-off of algorithm Approx in the quality of solutions is acceptable.

4.3 Impact of the Size of Social Networks

We investigate in this experiment the scalability of these three algorithms by increasing the number of nodes in the social network from 30 to 300. The precise numbers of nodes are 30, 43, 68, 106, 186 and 300. The number of required skills for each project is fixed at 6; the communication load constraint is set to 3. We can see in Fig. 4(a) that algorithm Brute-Force needs over 10 min to solve the team formation problem in the social network with 43 nodes, and has an explosive increase in execution time when the number of nodes is larger than 50. Although being able to deal with the team formation problem in the social network with 150 nodes within 10 min, algorithm Opt is still not scalable enough to handle social networks with more nodes. Due to employing approximation in the hierarchy establishment phase, algorithm Approx is able to handle the team formation problem on the social network with 300 nodes within 60 seconds. Although algorithm Approx cannot find the best team, as shown in Fig. 4(b), the cost difference ratio is smaller than 10 % when the number of nodes is smaller than or equal to 106. Experimental result shows that algorithm Approx is suitable for large social network with small trade-off in the quality of solutions.

5 Conclusion

In this paper, we proposed the team formation problem with the communication load constraint in expert social networks. To solve this problem, we presented a two-phase framework and developed three algorithms, namely algorithm Brute-Force, algorithm Opt and algorithm Approx. The experimental results showed that algorithm Opt is able to obtain optimal solutions efficiently by avoiding some teams being sent to the hierarchy establishment phase. Moreover, in the cases that nearly-optimal solutions are acceptable, algorithm Approx is much more scalable than algorithm Opt in large social networks at a cost of small increase in communication cost.

References

1. Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A., Leonardi, S.: Power in unity: forming teams in large-scale community systems. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management (2010)
2. Baykasoglu, A., Dereli, T., Das, S.: Project team selection using fuzzy optimization approach. Cybern. Syst. **38**(2), 155–185 (2007)
3. Chen, S.J., Lin, L.: Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering. IEEE Trans. Eng. Manage. **51**(2), 111–124 (2004)
4. Fitzpatrick, E.L., Askin, R.G.: Forming effective worker teams with multi-functional skill requirements. Comput. Ind. Eng. **48**(3), 593–608 (2005)
5. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman & Co., San Francisco (1979)
6. Gaston, M.E., Simmons, J., des Jardins, M.: Adapting network structure for efficient team formation. In: Proceedings of the AAAI 2004 Fall Symposium on Artificial Multi-Agent Learning (2004)
7. Juang, M.C., Huang, C.C., Huang, J.L.: Efficient algorithms for team formation with a leader in social networks. J. Supercomput. **66**, 721–737 (2013)
8. Kargar, M., An, A.: Discovering top-k teams of experts with/without a leader in social networks. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management (2011)
9. Lappas, T., Liu, K., Terzi, E.: Finding a team of experts in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2009)
10. Singh, S., Srivastava, R., Kumar, V., Agarwal, S.: An approximate algorithm for degree constraint minimum spanning tree. In: International Conference on Computer and Communication Technology (2010)
11. Zakarian, A., Kusiak, A.: Forming teams: an analytical approach. IIE Trans. **31**(1), 85–97 (1999)

Information Diffusion Pattern Mining over Online Social Media

Eric Hsueh-Chan Lu[1(✉)] and Hui-Ju Hung[2]

[1] Department of CSIE, National Taitung University, Taitung, Taiwan
luhc@nttu.edu.tw
[2] Academia Sinica, Taipei, Taiwan
hjhung4@iis.sinica.edu.tw

Abstract. With the rapid development of Web 2.0 technology, online social media have become increasingly popular and influential. In online social media, such as Reddit, Digg, Twitter and Weibo, users can post, vote and comment posted stories and other users' comments. Users, together with story and corresponding feedbacks, form a heterogeneous information diffusion network. To analyze how information diffuses among different users, we need to better understand a few key factors, including (1) frequent appearing sub-structures, called motifs, in the network and (2) evolution of a motif. In this paper, we explore the *MOtif-based Sequential Pattern* (*MOSP*) to facilitate the understanding of motif evolution along the time. Furthermore, we propose *Topological MOSP* (*T-MOSP*) and *Propagative MOSP* (*P-MOSP*) to observe frequent sequence of motifs in different angles. Facing a large volume of graph data, Motif mining is time-consuming. Therefore, we devise efficient mining algorithms, namely, *Motif-Mine* and *Lattice-based Temporal Sequential Pattern Mine* (*LTSP-Mine*), to discover motifs and sequences of motifs, respectively. Extensive experimental evaluation on Digg demonstrates that *T-MOSP* and *P-MOSP* discovered by the proposed algorithms can efficiently and effectively capture and summarize the information diffusion patterns in online social media.

Keywords: Information diffusion pattern · Online social media · Heterogeneous network · Motif-based Sequential Pattern

1 Introduction

In this work, we propose to study the patterns of information propagation over online social media, such as Reddit, Digg and Twitter. In previous studies, we've known that patterns of interconnections occurring in complex network are not random. The structure of design principles among those interconnections in complex network have been studied in cell phone [15], instant message [7], blog [9, 17], Flickr [2], email [14], and protein interaction [1] networks. Nevertheless, it is still unknown that how people interact with each other and how information flows among users in online social media. To understand the patterns of information flow in online social media is not only interesting as a scientific finding, but also beneficial for other applications or research fields. For example, as we known, social network data is not always available and it is challenging to obtain and release a real large scale social network dataset. Therefore, it

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 137–148, 2014.
DOI: 10.1007/978-3-319-13186-3_13

is very useful if there is a network generator which helps generate synthetic data for research purpose. Thus, our research can facilitate to develop network/graph generator which can well synthesize the complex network such as online social media. In addition, there are a lot of efforts in performing popularity prediction for a given story in online social media [3, 4]. This technique helps website to strategically launch advertisements and maximize profits. Our research not only can help people to understand how information propagate over online social network, but can be potentially beneficial for popularity prediction and useful for the viral marking strategy design.

In those online social media, users post interesting stories/news; other users may vote/comment/retweet those stories. So the information is flowed among users in this way. For a story, the information propagation can be represented as a heterogeneous network. Without loss of generality, let us consider Digg as an example. Digg contains 3 types of objects, namely user, story and comment. Links between users and stories are generated by actions such as submit, digg and comment; and the edges between comments/stories shows the relationship of "reply". Every link is associated with an action time. As there are multi-type of objects (nodes) and the actions linking objects may have different semantic meaning and is also directional, we represent the data collected from Digg as a (directed) heterogeneous network as shown in Fig. 1.

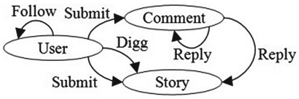


Fig. 1. Schema for digg social network.

To understand the pattern of information propagation over online social media, we turn to the tools of frequent pattern mining over a graph database. Given a collection of information propagation graphs, and each graph corresponds to one story information, we aim to discover the frequent information propagating patterns which are called frequent motifs. Frequent motifs reflect the frequent structures of social information propagations. Though frequent motifs have been investigated in other type of complex networks, relationship among motifs in the temporal dimension has never been explored. Exploring the temporal relationships among motifs is interesting and worthy to study. Such pattern is called as *MOtif-based Sequential Pattern* (*MOSP*). We observe that two kinds of *MOSPs* are interesting and insightful as follows:

1. *Topological MOSP* (*T-MOSP*)*: T-MOSP* is designed to observe that how the social group enlarges according to the time. For two consecutive motifs of *T-MOSP*, the latter one contains the former one. Hence, *T-MOSP* reflects the frequent pattern enlarging situations of social group including social users and social actions.
2. *Propagative MOSP* (*P-MOSP*)*: P-MOSP* is designed as to observe that how the information propagates according to the time. For two consecutive motifs of *P-MOSP*, there must be at least one directed link and one overlapped node between them. Hence, *P-MOSP* reflects the information propagations of social network.

In this paper, we propose new algorithms named *Motif-Mine* and *Lattice-based Temporal Sequential Pattern Mine* (*LTSP-Mine*) to discover frequent motifs and both of *T-MOSP* and *P-MOSP*, respectively. As introduced before, the graph consists of heterogeneous types of nodes, thus it is even challenging to devise an efficient motif mining algorithm, as we need to handle the isomorphism checking for heterogeneous graph, which has not been studied before. Furthermore, to find the evolution of motif, we need to transform a heterogeneous graph to a frequent motif sequence. The length of sequence may be very large. For a frequent motif, all its sub-structures are all frequent motifs. Hence, a lattice structure is used to maintain numerous motif relationships and to reduce the computational cost for discovering the evolution of motif. The major contributions of this paper are list as follows.

1. This is a first attempt to investigate patterns of information flow over online social media. This study not only provides interesting scientific findings, but also is beneficial viral marketing, popularity prediction and graph generation.
2. In addition of mining frequent motif, we also investigate how motif evolves over the online social media, and define temporal information propagation patterns, namely *MOtif-based Sequential Pattern* (*MOSP*).
3. We devise an efficient algorithms *Motif-Mine* and *Lattice-based Temporal Sequential Pattern Mine* (*LTSP-Mine*) to discover frequent motifs and both *T-MOSP* and *P-MOSP*, respectively. The main concept is to reduce the number of candidate motif generations and maintain the relationship of motifs for improving the efficiency.
4. Finally, we conduct comprehensive experiments using a real social media dataset *Digg* to evaluate the performance of our work. The results show insightful motif evolution over the time and interesting findings.

The remainder of this paper is organized as follows. We briefly review the related work in Sect. 2. In Sect. 3, we formulate the problem. In Sect. 4, we first introduce the proposed algorithms *Motif-Mine* and *LTSP-Mine*. In Sect. 5, we perform an empirical performance evaluation. Finally, in Sect. 6, we summarize our conclusions and future work.

2 Related Work

To our best knowledge, this is the first comprehensive study to make use of the temporal annotations of the information propagation in online social media to investigate the patterns of information propagation in online social media. Specifically, a novel type of substructure is proposed – motif-based sequential pattern – that characterized how information is propagated in the online social media from temporal perspectives. The recent availability of large amount of data from a variety of networks (e.g., people/animals, user generated content, proteins and so on) has enabled the analysis of the structural or topological properties of different types of the networks. Research work has been done to analyze cell phone [15], instant message [7], blog [9, 17], Flickr [2], email [14], and protein interaction [1] networks. Those studies help understand the frequent building blocks/models of the networks, by analyzing the structural/topological properties of the networks. Nevertheless, those previous approaches have typically ignored the temporal attributes, thus cannot discover the local structural behavior patterns and their evolution, which are important for understanding the core principles of collective structural pattern over all the networks. In addition, instead of interested in the conventional network dataset, we propose to study information diffusion patterns over online social media such as Digg, Twitter and Reddit, which are emerging and popular social media.

The dynamics properties of large scale social networks have been studied extensively [5, 6, 13, 19, 20]. Yang and Counts discuss the information diffusion on Twitter. They study three major properties including speed, scale and range. However, there is no work discusses that how information propagates and shows the evolution pattern about the information diffusion. Along these lines, research has been carried out on studying information cascading triggered by specific events [8]. Note that our work is a general approach that characterizes different types of events/stories and the discovered information diffusion patterns capture how information is propagated over the online social media from the temporal perspective.

Information diffusion pattern discovery can be considered as frequent pattern mining in graph database. Many algorithms have been proposed to find the frequent patterns or motifs from a set of graphs which is called graph-transactions. In [11], Inokuchi *et al.*, first propose an Apriori-based algorithm named *AGM* to discover the frequent appearing sub-structures in a given graph data set. Kuramochi and Karypis propose *FSG* [12] for finding all frequent sub-graphs in large graph databases. In [18], Yan and Han propose *gSpan* to discover frequent sub-structures without candidate generation. Each graph is mapped to a unique DFS code based on its canonical label. Based on the lexicographic order of DFS code, gSpan can efficiently mine frequent sub-graphs by the DFS search strategy. In [10], Huan *et al.*, propose *FFSM* to reduce the number of redundant candidates while mining frequent sub-graphs. However, mining complete frequent patterns from graph databases is challenging since the operations, such as candidate-joining and sub-graph checking, generally are computational costly. *SPIN* [16] and *MARGIN* [17] are proposed to apply the concept of maximal patterns in sub-graph mining. However, in online social media context, the information diffusion pattern mining imposes new research challenges.

3 Problem Statement

In this section, we provide the formal definitions regarding the frequent motif and topological/propagative *MOSP* in heterogeneous graphs. Given a heterogeneous graph database $D = \{G_1, G_2, \ldots, G_{|D|}\}$ containing a set of heterogeneous graphs, each graph G can be represented as a series of social actions $R = \{r_1, r_2, \ldots, r_{|G|}\}$ which is called *actionset* corresponding to some specific stories. To help better understanding of those definitions, we also introduce auxiliary definitions such as social action, valid graph, graph isomorphism, and sub-graph.

Definition 1. Social Action: A social action is represented as a 5-tuple $r_i = <id_i, t_i, u_i, a_i, o_i>$, where id_i denotes the action id; t_i, denotes the delay time between the time of action r_i and the time when the story was submitted; u_i denotes the user who executes the action; a_i represents the action type which includes submit, digg and comment; and o_i represents the action object which may be empty or target action id.

Definition 2. Valid Graph: A heterogeneous graph G is called a valid graph if the corresponding actionset $R = \{r_1, r_2, \ldots, r_{|G|}\}$ satisfies the condition: For each $r \in R$, except the first r, there always can find an $r' \in R$ such that $r.o = r'.id$. In other words, a valid graph must be connected.

Definition 3. Graph Isomorphism: Given two valid graphs G_i and G_j, G_i and G_j are graph isomorphism, denoted as $G_i = G_j$, if their corresponding actionsets R_i and R_j satisfy the conditions: there exists a one-to-one mapping between R_i and R_j, g: $R_i \rightarrow R_j$, such that (1) $r_x.a = g(r_x).a$, (2) $r_y.a = g(r_y).a$, and (3) $g(r_x).id = g(r_y).o$, where $g(r_x)$, $g(r_y) \in R_j$, for any pairs of actions $r_x = <id_x, t_x, u_x, a_x, o_x>$, $r_y = <id_y, t_y, u_y, a_y, o_y> \in R_i$ and $id_x = o_y$.

Definition 4. Sub-Graph: Given two valid graphs G_i and G_j, G_i is called G_j's subgraph, denoted as $G_i \subseteq G_j$, if there exists at least one subgraph G_j' of G_j such that $G_i = G_j$'. In addition, G_j is called G_i's super-graph, denoted as $G_j \supseteq G_i$.

Definition 5. Motif: Given a heterogeneous graph database $D = \{G_1, G_2, \ldots, G_{|D|}\}$. A valid graph m can be represented as a motif if there is at least one heterogeneous graph G in D such that $G \supseteq m$. Note that motifs only represent the structures of information flows but who submitted the story/comment. Furthermore, the support of m, denoted as $sup(m)$, is defined as the number of graphs in D that are m's super-graphs; and the length (size) of m, denoted as $len(m)$, is defined as the number of submission (submit or comment) actions. m is also called a k-motif if $len(m) = k$.

Definition 6. Frequent Motif: Given a heterogeneous graph database $D = \{G_1, G_2, \ldots, G_{|D|}\}$ and a minimal support threshold θ. A k-motif m is called a frequent motif if $sup(m) \geq \theta$.

Definition 7. MOtif-based Sequential Pattern (MOSP): Given a heterogeneous graph database $D = \{G_1, G_2, \ldots, G_{|D|}\}$ and the corresponding frequent motifs $M = \{m_1, m_2, \ldots, m_{|M|}\}$ from D. For each G in D, a maximal subset M' of M can be extracted such that $G \supseteq m'$, $\forall m' \in M'$. Hence, all of the graph G_i in D can be transformed to a set of frequent motifs M'_i, i.e., $D = \{M'_1, M'_2, \ldots, M'_{|D|}\}$. A permutation P of subset of M is called a *MOSP* if there exists at least one M' in D such that $M' \supseteq P$. Furthermore, the support of P, denoted as $sup(P)$, is defined as the number of graphs in D that contain P.

Definition 8. Topological MOSP (T-MOSP): A *MOSP* $P = <m_1, m_2, \ldots, m_{|P|}>$ is called a *T-MOSP* if any consecutive pair of motifs in P exists containing relationship, i.e., $m_{k+1} \supseteq m_k$, $\forall\ 1 \leq k \leq |P|$-1.

Definition 9. Propagative MOSP (P-MOSP): A *MOSP* $P = <m_1, m_2, \ldots, m_{|P|}>$ is called a *P-MOSP* if any consecutive pair of motifs in P exists contacting relationship, i.e., m_{k+1} and m_k have at least one directed social action link, $\forall\ 1 \leq k \leq |P|$-1.

Problem Definition: Given a heterogeneous graph database D and a minimal support threshold θ. The problem is to find all frequent motifs, *T-MOSP* and *P-MOSP* in D.

4 Proposed Method

To understand the patterns of information propagation over online social media, there are two problems need to be solved:

1. *How to obtain the frequent motifs in a heterogeneous graph database?* Although the concept of frequent motif mining from graphs has been proposed in previous literatures, most of them focus on the problem of frequent motif mining in homogeneous graphs. According to the previous studies related to frequent motif finding [20], there are four common motif structures including Chain, Star, Loop and PingPong have been defined. An intuitive solution is that the motifs are directly used based on the previous observations. However, nobody can guarantee that there is no other motif structure in our graph database, especially in a heterogeneous one. Therefore, an automatic heterogeneous frequent motif finding algorithm is highly desired. In this section, we describe the proposed algorithm *Motif-Mine* to efficiently discover all the frequent motifs from the heterogeneous graphs.
2. *How to obtain T-MOSP and P-MOSP after obtaining the frequent motifs?* An intuitive solution is that we first find all the *MOSPs* and then extract the *T-MOSP* and *P-MOSP* based on their constraints. However, a lot of *MOSPs* discovered by existed sequential pattern mining approaches are not *T-MOSPs* neither *P-MOSPs*. Hence, the computation performance is very inefficient. More specifically, based on our definitions of *T-MOSP* and *P-MOSP*, motifs become a pattern must satisfy some constraints. For example, two motifs can become a *T-MOSP* or a *P-MOSP* if the later one contains the former one or the later one connected to the former one, respectively. Therefore, how to reduce the unnecessary computations is a key point to improve the efficiency. In this section, we describe the proposed algorithm *Lattice-based Temporal Sequential Pattern Mine* (*LTSP-Mine*) to efficiently discover all *T-MOSPs* and *P-MOSPs* from the frequent motifs mined by *Motif-Mine*.

4.1 Motif-Mine

To obtain heterogeneous frequent motifs, we propose an Apriori-like algorithm *Motif-Mine* here. The concept of *Motif-Mine* is to generate the entire candidate (k+1)-motifs from all the frequent k-motifs and check which candidate motifs are frequent motifs by counting their frequencies. By adopting *Motif-Mine*, we can obtain all the frequent motifs in the graph database. *Motif-Mine* can be divided into three steps.

1. *Frequent 2-Motifs Finding*. We enumerate all possible 2-motifs and check which 2-motifs are frequent. As shown in Fig. 2, the number of possible 2-motifs is only 2, i.e., m_1 and m_2. Hence, we can directly check whether m_1 and m_2 are frequent without generating frequent 1-motifs.
2. *Candidate Motif Generation*. The traditional approach for candidate motif generation may miss some potential motifs. Take Fig. 2 as an example, m_1 and m_2 are 2-motifs and m_3, m_4 and m_5 are 3-motifs. m_3 is generated from m_1 and m_2. However, m_4 is generated from two m_1. It is very different from the traditional candidate

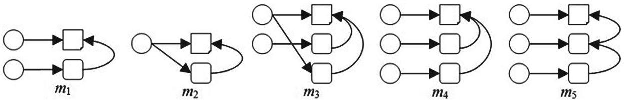


Fig. 2. Five motifs.

join process. Therefore, *Motif-Mine* uses a "bottom up" approach, where candidate frequent motif is generated by extending one more social action based on existing frequent motif. In other words, we extend all possible candidate *k*-motifs from a frequent (*k*-1)-motif by adding one possible social action. In Fig. 2, all possible candidate 3-motifs generated from m_1 are shown in Fig. 3. There are 6 kinds of possible candidate 3-motifs generated from m_1. The candidate generation process terminates when no further frequent motifs can be extended.

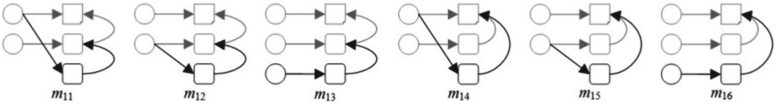


Fig. 3. All possible candidate 3-motifs generated from m_1.

3. *Frequent Motif Determination*. To determine whether a candidate motif is frequent, we need to count the number of stories contain the candidate motif. It is inefficient if we directly scan all stories to determine which candidate motifs are frequent, since the computation complexity of sub-graph isomorphism checking is extremely high. To reduce the number of graph isomorphism checks, before scanning through all stories to obtain the support of a candidate *k*-motif, we will first check whether all of the (*k*-1)-sub-motifs of the candidate *k*-motif are frequent according to downward closure lemma. *Motif-Mine* can save the computation cost for scanning all stories if a candidate motif contains any infrequent sub-motif. Besides, those (*k*-1)-sub-motifs, which are invalid graphs, are ignorable as well. Take m_{13} in Fig. 3 as an example, there are 3 2-sub-motifs as shown in Fig. 4. However, we only need to check whether m_{133} is frequent since both of m_{131} and m_{132} are invalid. Furthermore, m_{13} can be ignored if m_{133} is not frequent. Actually, the frequency checking procedure of a candidate *k*-motif cannot be avoided if there is only one valid (*k*-1)-sub-motif, because any candidate *k*-motif must be extended from a frequent (*k*-1)-motif. Hence, the (*k*-1)-sub-motif must be frequent when a candidate *k*-motif only has one valid (*k*-1)-sub-motif. For example, we can count the support of m_{13} by scanning all stories in the database directly, since it only has one valid (*k*-1)-sub-motif, i.e., m_{133}.

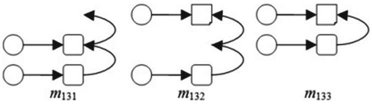


Fig. 4. All 2-sub-motifs of m_{13}.

Table 1. Motif sequences

S_{id}	Motif sequence
S_1	*2C*: (S_1, C_{11}), *2C*: (S_1, C_{12}), *2C*: (C_{12}, C_{13}), *3C*: (S_1, C_{12}, C_{13}), *3S*: (S_1, C_{11}, C_{12}), *4CS*: (S_1, C_{11}, C_{12}, C_{13})
S_2	*2C*: (S_2, C_{21}), *2C*: (S_2, C_{22}), *2C*: (C_{21}, C_{23}), *3C*: (S_2, C_{21}, C_{23}), *3S*: (S_2, C_{21}, C_{22}), *4CS*: (S_2, C_{21}, C_{22}, C_{23})
S_3	*2C*: (S_3, C_{31}), *2C*: (S_3, C_{32}), *2C*: (C_{32}, C_{33}), *3S*: (S_3, C_{31}, C_{32})

4.2 LTSP-Mine

After obtaining the frequent motifs, the next task is to mine the *T-MOSPs* and *P-MOSPs*. We first transform the story database to the motif sequence database. Table 1 shows three motif sequences based on the frequent motifs. Each element in the motif sequences consists of a motif id and a set of corresponding social actions. For example, "2*C*: (S_1, C_{11})" in S_1 represents there is a 2-Chain motif in S_1 and this motif consists of two social actions S_1 and C_{11}. To discover the patterns, an intuitive solution is that we first find all *MOSPs* based on the existed sequential pattern mining algorithm and filter out some of *MOSPs*, which are not *T-MOSPs* or *P-MOSPs*. However, it is very inefficient since too many redundant *MOSPs* are found. For example, a story data may contain several 2-Chain motifs if many users comment each other. A lot of *MOSPs* "2-Chain → 2-Chain" are discovered. How to reduce the number of redundant *MOSP* generations in the pattern mining phase is a key point to improve the mining efficiency.

Based on the definitions of *T-MOSP* and *P-MOSP*, two motifs can be combined as a pattern if there exists the containing and contacting relationships between the motifs, respectively. Therefore, we need to check the containing and contacting relationships among the motifs in the pattern mining phase. However, the computation cost of motif relationship checking is high. It is not efficient if we check the motif relationships in every candidate generations. Hence, we propose a lattice structure to retain the containing and contacting relationships among those motifs. More specifically, before mining the patterns, all the frequent motifs are built as a lattice structure. Actually, the lattice structure is also established while *Motif-Mine* is finished. Two frequent motifs will be connected if one of them is generated from another. The reason is that a frequent *k*-motif must be generated from a frequent (*k*-1)-motif in our design.

After building the lattice structure, we can mine all the patterns. Any *T-MOSP* or *P-MOSP* consists of at least two motifs. For a *T-MOSP*, two motifs can be formed as a pattern if the later one contains the former one. Take S_1 in Table 1 as an example, the patterns "*2C* → *3C*" and "*2C* → *3S*" are *T-MOSPs* since *3C* and *3S* contain *2C* according to the lattice structure, and the social actions of *3C* and *3S*, i.e.,{S_1, C_{12}, C_{13}} and {S_1, C_{11}, C_{12}}, also contain the social actions of 2*C*, i.e., {S_1, C_{12}}. For a P-*MOSP*, two motifs can be joined as a pattern if the later one contacts the former one. For example, the pattern "2*C* → 2*C*" is a *P-MOSP* in S_1 since the social actions of the later "2*C*", i.e., {C_{12}, C_{13}}, contact the social actions of the former "2*C*", i.e., {S_1, C_{12}}. They can be connected by the social action "C_{12}". Finally, we can obtain all the *T-MOSPs* and *P-MOSPs* according to the lattice structure and social actions.

5 Experimental Evaluations

In order to conduct empirical study on the characteristics of the frequent motifs, T-*MOSPs* and *P-MOSPs* in online social media, we collect data from Digg. Experiments can be divided into two parts, (1) findings regarding the frequent motifs and (2) findings of *T-MOSP* and *P-MOSP*. All of the experiments were implemented in Visual Studio 2010 on an Intel Xeon CPU E7-4870 2.40 GHz (4 processors) machine with 128 GB of memory running Microsoft Windows Server Enterprise.

5.1 Experimental Dataset

We collect data from a real social media, which was collected in the Digg social website. The Digg social media provides a web API service that allows users to collect Digg data. We collect 21,004 stories among 60,415 unique users during the period of August 13, 2010 and January 26, 2012. All of the digg stories can be categorized into 10 topics including Business, Entertainment, Gaming, Lifestyle, Offbeat, Politics, Science, Sports, Technology, and World News. A story can be digged or commented by any user, and a comment can be commented by another user as well. The average numbers of diggs and comments of a story are 40 and 6, respectively.

5.2 Discussion of Frequent Motifs

In this sub-section, we show and discuss the motifs mined from 10 various topic of Digg dataset when the minimal support threshold is set as 2 %. Note that in the experiment, we confirm that there exist motifs (namely, *common motifs*) which were discovered in previous studies [20]; we also discover *new motifs*, which are not presented in previous works.

Common Motifs. Previous works [20] have proposed four kinds of common frequent motifs including n-Chain, n-Star, PingPong and n-Loop, where n indicates the length of motif. Figure 5 shows the support percentage results, which is calculated as the support count of motif divided by the number of graphs, of n-Chain, PingPong and n-Loop under various topics of datasets. Here, we do not put the support percentage results of n-Star since such motif type occurs almost in every story. Such phenomenon is easy to understand because the n-Star is easy to occur when several users reply the same story or comment. For the remaining motif types, we observe that the support percentages of n-Chain, PingPong and n-Loop for the topic “Politics” are significantly higher than those of any other topics. The main reason is that the average number of comments of a politics story is significantly more than that of other topical stories. Hence, it is more likely that those motifs are generated from political story data. We observe that besides of politics, world news in general has more number of comments than other topics. Therefore, similar to the political topic, we discover relative more Chain motifs in stories of world news comparing to other topics as well.

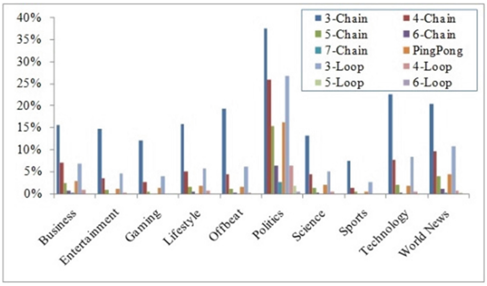
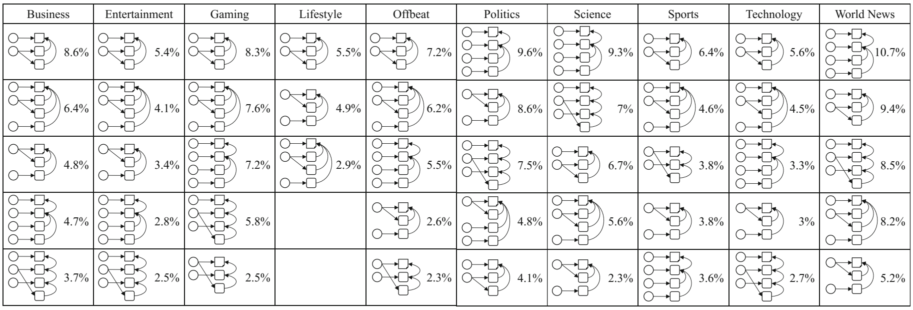


Fig. 5. Support percentages of n-Chain, PingPong and n-Loop.

New Motifs. In addition of those motifs, we also discover new motifs, which are not presented in previous works. Figure 6 shows the 5 most frequent new motifs from 10 various topics of datasets. The support is converted into a percentage, based on the ratio of frequency to the total number of stories in the dataset. Hence, the results across datasets are comparable. We can clearly observe that most of top-1 new motif structures across different topics are always similar to Star structures. The reason is that Star structure is the most common motif existing in complex network, those new motifs can be considered as variants of Start structure, by incorporating heterogeneous nodes into consideration. Besides, we observe that the Politics, Science and World News datasets contains more large motifs and PingPong-like structures than other topics of datasets. This phenomenon may be explained by these topics are news-based stories. The conversation or discussion behaviors may be easier to form large and PingPong-like motif structures in such topics of stories.



Business	Entertainment	Gaming	Lifestyle	Offbeat	Politics	Science	Sports	Technology	World News
8.6%	5.4%	8.3%	5.5%	7.2%	9.6%	9.3%	6.4%	5.6%	10.7%
6.4%	4.1%	7.6%	4.9%	6.2%	8.6%	7%	4.6%	4.5%	9.4%
4.8%	3.4%	7.2%	2.9%	5.5%	7.5%	6.7%	3.8%	3.3%	8.5%
4.7%	2.8%	5.8%		2.6%	4.8%	5.6%	3.8%	3%	8.2%
3.7%	2.5%	2.5%		2.3%	4.1%	2.3%	3.6%	2.7%	5.2%

Fig. 6. New motifs.

5.3 Discussion of T-MOSP and P-MOSP

In this sub-section, we examine the *T-MOSPs* and *P-MOSPs* we discovered through our proposed algorithms. Figure 7 shows the top 4 *T-MOSPs* and *P-MOSPs*. We can observe that the probability of Star motif evolution is significantly higher than that of Chain motif evolution in every topic. This indicates that the users are usually only interested in the submitted story. Hence, they have a higher probability to comment the story then comment another user's comment. However, we still find some interesting phenomena in this experiment: (1) The Politics, Science and World News datasets contain more Chain-like motif evolution than other topic of datasets. This indicates news-based stories have higher probabilities to form a series of talks. (2) The Gaming dataset has high percentage in the Star-like motif evolution. We also observe that the special *P-MOSPs* under various topic datasets. For example, the Politics, Science, Technology and World News dataset have higher probabilities to generate long motif evolutions such as "2C→2S→2S".

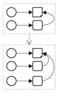
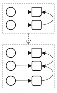
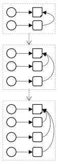
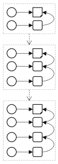
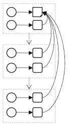
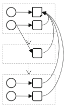
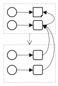
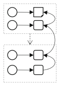
	2C → 3S	2C → 3C	2C → 3S → 4S	2C → 3C → 4C		2C → 2S → 2S	2C → 1S' → 2S	2C → 2S	2C → 2C
T-MOSPs					P-MOSPs				
Business	21.5%	6.4%	13%	1.5%	Business	2.6%	1.7%	1.3%	1.5%
Entertainment	30.1%	8.7%	17.6%	1.2%	Entertainment	3.7%	1.3%	1%	1.2%
Gaming	55%	21.6%	37.7%	3.4%	Gaming	1.2%	0.7%	4.2%	3.4%
Lifestyle	18%	1.6%	8.8%	2.8%	Lifestyle	1.3%	0.9%	3.3%	2.8%
Offbeat	49.3%	19%	33.9%	3%	Offbeat	1.2%	0.8%	3.8%	3%
Politics	22.4%	11.7%	13.7%	3.7%	Politics	5.9%	1.6%	3.7%	3.7%
Science	50.5%	20.2%	29.8%	4.7%	Science	5.3%	1.8%	4.7%	4.7%
Sports	44.5%	15%	27.7%	1.7%	Sports	3.6%	1.8%	1.4%	1.7%
Technology	29%	9.9%	17%	2.2%	Technology	3.9%	1.5%	1.4%	2.2%
World News	21.9%	8.2%	12%	1.3%	World News	4.5%	1.5%	3.4%	1.3%

Fig. 7. Top T-MOSPs and P-MOSPs with their appearing probability in different categories.

6 Conclusions and Future Work

In this paper, we have proposed a novel framework and two kinds of *MOtif-based Sequential Patterns* (*MOSP*) including *Temporal MOSP* (*T-MOSP*) and *Propagative MOSP* (*P-MOSP*) to understand the patterns of information propagation over online social media. Furthermore, we have proposed not only an efficient algorithm, namely *Motif-Mine*, to find all the frequent motifs in a heterogeneous social network, but also an algorithm, namely *Lattice-based Temporal Sequential Pattern Mine* (*LTSP-Mine*), to discover all the *MOSPs* among motifs. To the best of our knowledge, this is the first work on efficiently mining heterogeneous frequent motifs and revealing the motif evolution in heterogeneous networks.

A comprehensive experimental study has been conducted based on the data collected from a representative online social media - Digg. In the experiment, we not only discover those common existed motifs, but also uncover the new motifs, which are unique over online social media due to the heterogeneity of the graph. In addition, we also observe that information flow patterns for different topics of stories exhibit different behaviors, e.g., "Politics" stories show significantly higher probability to have *n*-Chain motif for information flow. Although this study is experimented over the dataset collected from Digg, the research problem and developed research methodologies are general enough and can be applied to other online social media. Therefore, we plan to further our study by collecting data from other online social media such as Twitter and Reddit to investigate the patterns of information flow.

References

1. Alon, U.: Network motifs: theory and experimental approaches. Nat. Rev. Genet. **8**(6), 450–461 (2007)
2. Anagnostopoulos, A., Kumar, R., Mahdian, M.: Influence and correlation in social networks. In: KDD, pp. 7–15 (2008)
3. Asur, S., Huberman, B.: Predicting the future with social media. In: Web Intelligence, pp. 492–499 (2010)
4. Bakshy, E., Hofman, J., Mason, W., Watts, D.: Everyone's an influencer: quantifying influence on twitter. In: WSDM, pp. 65–74 (2011)
5. Chen, B., Zhao, Q., Sun, B., Mitra, P.: Predicting blogging behavior using temporal and social networks. In: ICDM, pp. 439–444 (2007)
6. Choudhury, M., Sundaram, H., John, A., Seligmann, D.: Dynamic prediction of communication flow using social context. In: Hypertext, pp. 49–54 (2008)
7. Du, N., Faloutsos, C., Wang, B., Akoglu, L.: Large human communication networks: patterns and a utility-driven generator. In: KDD, pp. 269–278 (2009)
8. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. In: WWW, pp. 491–501 (2004)
9. Guo, L., Tan, E., Chen, S., Zhang, X., Zhao, Y.: Analyzing patterns of user content generation in online social networks. In: KDD, pp. 369–378 (2009)
10. Huan, J., Wang, W., Prins, J.: Efficient mining of frequent subgraph in the presence of isomorphism. In: ICDM, pp. 549–552 (2003)
11. Inokuchi, A., Washio, T., Motoda, H.: An apriori-based algorithm for mining frequent substructures from graph data. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 13–23. Springer, Heidelberg (2000)
12. Kuramochi, M., Karypis, G.: An efficient algorithm for discovering frequent subgraphs. TKDE **16**, 1038–1051 (2004)
13. Lin, Y., Chi, Y., Zhu, S., Sundaram, H., Tseng, B.: Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In: WWW, pp. 685–694 (2008)
14. Malmgren, R., Hofman, J., Amaral, L., Watts, D.: Characterizing individual communication patterns. In: KDD, pp. 607–616 (2009)
15. Nanavati, A., Singh, R., Chakraborty, D., Dasgupta, K., Mukherjea, S., Das, G., Gurumurthy, S., Joshi, A.: Analyzing the structure and evolution of massive telecom graphs. IEEE TKDE **20**, 703–718 (2008)
16. Prins, J., Yang, J., Huan, J., Wang, W.: Spin: mining maximal frequent subgraphs from graph databases. In: KDD, pp. 581–586 (2004)
17. Thomas, L., Valluri, S., Karlapalem, K.: Margin: maximal frequent subgraph mining. In: ICDM, pp. 1097–1101 (2006)
18. Yan, X., Han, J.: gSpan: graph-based substructure pattern mining. In: ICDM, pp. 721–724 (2002)
19. Yang, J., Counts, S.: Predicting the speed, scale, and range of information diffusion in twitter. In: ICWSM (2010)
20. Zhao, Q., Tian, Y., He, Q., Oliver, N., Jin, R., Lee, W.-C.: Communication motifs: a tool to characterize social communications. In: CIKM, pp. 1645–1648 (2010)

Mining Frequent Progressive Usage Patterns Across Multiple Mobile Broadcasting Channels

Bijay Prasad Jaysawal(✉) and Jen-Wei Huang

National Cheng Kung University, Tainan, Taiwan
bijay@jaysawal.com.np, jwhuang@mail.ncku.edu.tw

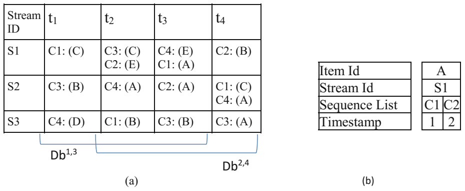
Abstract. Sequential pattern mining is to find frequent data sequences with time. When sequential patterns are generated, the newly arriving patterns may not be identified as frequent sequential patterns due to the existence of old data and sequences. Progressive sequential pattern mining aims to find most up-to-date sequential patterns given that obsolete items will be deleted from the sequences. When sequences come with multiple data streams, it is difficult to maintain and update the current sequential patterns. Even worse, when we consider the sequences across multiple streams, previous methods could not efficiently compute the frequent sequential patterns. In this work, we propose an efficient algorithm PAMS to address this problem. PAMS uses a PSM-tree to insert new items, update current items, and delete obsolete items. The experimental results show that PAMS significantly outperforms previous algorithms for mining progressive sequential patterns across multiple streams.

Keywords: Progressive mining · Sequential pattern · Multiple data streams

1 Introduction

Mobile broadcasting channels broadcast some contents while many users can watch those contents from different broadcasting channels as per their interests. Figure 1(a) shows an example of multiple mobile broadcasting channels. $S1$, $S2$, and $S3$ represent three mobile broadcasting channels. Contents from these channels are watched at each time point t_i. $C1$, $C2$, $C3$, and $C4$ are different mobile users. A, B, C, D, and E represent different contents. The contents watched by users at different time become a sequence according to their time orders. At time t_1, user $C1$ watched (C) from $S1$, user $C3$ watched (B) from $S2$, and user $C4$ watched (D) from $S3$. At any point of time a user can watch only one content and from one channel. In this example, if the frequency of usage pattern is no less than a user defined minimum support, that usage pattern is called as frequent usage pattern. Period of Interest (POI) is a sliding window, whose length is a user-specified time interval. We use POI as 3 and minimum support as 0.5 for running example. In the period t_1 to t_3, both $C1$ and $C3$ have

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 149–155, 2014.
DOI: 10.1007/978-3-319-13186-3_14



Stream ID	t_1	t_2	t_3	t_4
S1	C1: (C)	C3: (C) C2: (E)	C4: (E) C1: (A)	C2: (B)
S2	C3: (B)	C4: (A)	C2: (A)	C1: (C) C4: (A)
S3	C4: (D)	C1: (B)	C3: (B)	C3: (A)

$Db^{1,3}$
$Db^{2,4}$
(a)

Item Id	A	
Stream Id	S1	
Sequence List	C1	C2
Timestamp	1	2

(b)

Fig. 1. (a) Example database (b) Internal node structure

the following usage pattern across multiple broadcast channels: $\langle C{-}S1\rangle\langle B{-}S3\rangle$, where element is in the form $\langle Content - Channel\rangle$. When the time moves to t_4, the POI contains t_2 to t_4. $C2$ and $C4$ have the usage pattern $\langle E - S1\rangle\langle A - S2\rangle$ across multiple broadcasting channels.

If we think channels as streams and usage patterns as sequential patterns, the problem of finding frequent progressive usage patterns can be expressed as progressive sequential patterns mining across multiple data streams. It would be also applicable to mobile user's temporal behavior patterns where location can be treated as streams and requested services as items by mobile users.

Many research have been done on sequential pattern mining in static databases and incremental databases. Recently, sequential pattern mining on progressive database has been started by [6], which deals with the addition of newly arriving data and the deletion of obsolete data in the database at the same time based on a user defined POI. Note that, in the progressive sequential pattern mining problem, the length of POI can be adjusted at any time.

Another challenging field is the data stream mining where algorithms must get the information or compute the summary in one pass. Due to the increasing need of real-time knowledge and the limitation of data streams, the newer data has more importance. In multiple data streams, the sequential pattern of a sequence can be obtained from same stream as well as across streams. Mining sequential pattern across multiple data streams has been addressed in [10].

To address the progressive mining of sequential pattern across multiple streams problem, we propose an algorithm PAMS. PAMS stands for Progressive sequential pattern mining Across Multiple Streams. PAMS maintains a PSM-tree to keep the information of the progressive database and up-to-date across streams sequential patterns in each POI.

The rest of this paper is organized as follows. Section 2 describes related works. The details of PAMS and PSM-tree are described in Sect. 3. The performance evaluation and comparisons with previous algorithm are given in Sect. 4. Finally, we conclude the paper in Sect. 5.

2 Related Works

Many research have been done to find frequent sequential pattern in static databases [1,2,7] and in incremental databases [4].

Later, the sequential pattern mining in data streams emerged. Many research have been done in sequential pattern mining in data streams [5,9] but very less research have been done in mining multiple data streams. In [8], authors have discussed about research issues in mining multiple data streams. MILE [3] finds sequential patterns in multiple streams but the problem definition of sequential pattern across streams is different than in [10].

Another interesting research area related to sequential pattern mining is started by [6] called sequential pattern mining in progressive database. Pisa in [6] dealt with the sequential pattern mining problem in a progressive database where new data are added and old data are removed over time.

The problem of mining sequential pattern across multiple data streams has been discussed in IAspam [10]. IAspam uses the sliding window, bitmap representation of current sliding window, and lexicographic tree for mining purpose.

3 Progressive Mining of Sequential Pattern Across Multiple Streams

3.1 PSM-tree

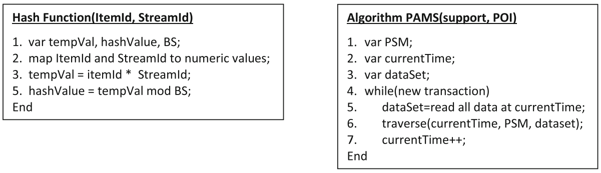
PAMS maintains and uses PSM-tree for mining purpose. PSM-tree contains all the information and helps PAMS to generate frequent sequential patterns in each POI. There are two types of nodes in PSM-tree: root node and internal nodes. The root node contains a hash table in which the entries contain references to children nodes directly under root. Each internal node stores the ItemID, StreamID, sequence list, and timestamp as shown in Fig. 1(b). The Sequence list is a list containing sequence IDs (in our example mobile user) accompanied by timestamp. Only the nodes in the first and the second levels have to maintain the corresponding timestamps. From third level, the timestamps for the sequence IDs are the same as the timestamps for the same sequence IDs in the second level. The path from root node to any other node represents the candidate sequential pattern appearing in that sequence across multiple streams.

In the PSM-tree, too many children nodes appear under the root node. Reasons are: different types of items in transactions, different streams generate different nodes for the same type of item. Too many children under root node slow down the search for existing node under root. To solve this issue, we have used hash table implementation in root node, which contains ItemID and StreamID as key, and the reference of the child node as value.

3.2 Algorithm PAMS

PAMS maintains the information of each sequence and each candidate sequential pattern progressively. PAMS uses PSM-tree to store all sequences from one POI to another. While processing at timestamp $t + 1$, PAMS traverses the PSM-tree of timestamp t in post order (children first, then the node itself) and updates the PSM-tree of timestamp t for timestamp $t + 1$. In the procedure of traversing

PSM-tree, PAMS does the following tasks: deletes the obsolete sequence IDs and nodes from the PSM-tree, updates current sequence list of nodes in the PSM-tree, and inserts newly arriving elements into the PSM-tree.



```
Hash Function(ItemId, StreamId)
var tempVal, hashValue, BS;
map ItemId and StreamId to numeric values;
tempVal = itemId *  StreamId;
hashValue = tempVal mod BS;
End
```

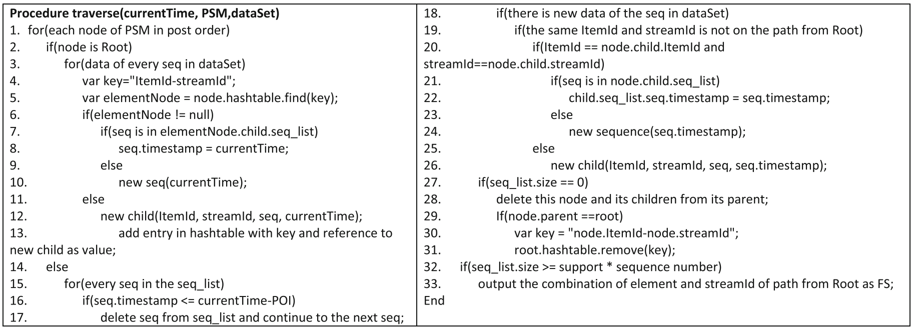
(a)

```
Algorithm PAMS(support, POI)
var PSM;
var currentTime;
var dataSet;
while(new transaction)
    dataSet=read all data at currentTime;
    traverse(currentTime, PSM, dataset);
    currentTime++;
End
```

(b)

Fig. 2. (a) Hash function (b) Algorithm PAMS

The detailed algorithm is shown in Fig. 2(b). PAMS gets the incoming data of all streams at current timestamp and traverses the PSM-tree. Then, PAMS moves forward to the next timestamp.



```
Procedure traverse(currentTime, PSM,dataSet)
1.  for(each node of PSM in post order)
2.      if(node is Root)
3.          for(data of every seq in dataSet)
4.              var key="ItemId-streamId";
5.              var elementNode = node.hashtable.find(key);
6.              if(elementNode != null)
7.                  if(seq is in elementNode.child.seq_list)
8.                      seq.timestamp = currentTime;
9.                  else
10.                     new seq(currentTime);
11.             else
12.                 new child(ItemId, streamId, seq, currentTime);
13.                     add entry in hashtable with key and reference to
new child as value;
14.     else
15.         for(every seq in the seq_list)
16.             if(seq.timestamp <= currentTime-POI)
17.                 delete seq from seq_list and continue to the next seq;
18.                 if(there is new data of the seq in dataSet)
19.                     if(the same ItemId and streamId is not on the path from Root)
20.                         if(ItemId == node.child.ItemId and
streamId==node.child.streamId)
21.                             if(seq is in node.child.seq_list)
22.                                 child.seq_list.seq.timestamp = seq.timestamp;
23.                             else
24.                                 new sequence(seq.timestamp);
25.                         else
26.                             new child(ItemId, streamId, seq, seq.timestamp);
27.             if(seq_list.size == 0)
28.                 delete this node and its children from its parent;
29.                 If(node.parent ==root)
30.                     var key = "node.ItemId-node.streamId";
31.                     root.hashtable.remove(key);
32.         if(seq_list.size >= support * sequence number)
33.             output the combination of element and streamId of path from Root as FS;
End
```

Fig. 3. Procedure traverse

The procedure traverse is shown in Fig. 3 which traverses PSM-tree in post order. The basic idea of the procedure traverse is to append each newly arriving element of all sequences into PSM-tree. For root node, PAMS examines all sequences which have new data. PAMS processes the root node in lines 2 to 13. If the node is root, PAMS checks every newly arriving data in the dataSet. Then the ItemID and StreamID are combined to be used as key to search in the hash table. If the element is found, PAMS uses the node reference to directly access the node. Then, PAMS checks for the sequence ID in the sequence list of that child node in lines 7 to 10. If the sequence ID is found, PAMS simply updates the timestamp of that sequence to the new timestamp. Otherwise, PAMS creates a new sequence and inserts into the sequence list of that child. But, If the element is not found in hash table, PAMS creates a new child node with the corresponding ItemID, StreamID, sequence ID, and timestamp in line 12. Also, PAMS adds an entry in the hash table for this new child in line 13.

For internal nodes, PAMS examines the sequence list of that node to find any newly arriving data of the sequences in the list. The details are shown from line 15 to line 33. PAMS first deletes the obsolete sequence IDs from the sequence list in lines 16 to 17. If there is no sequence ID left in the sequence list, PAMS prunes this node away from its parent and goes to next node as shown in lines 27 to 31. On the other hand, if there are still some sequence IDs left in the list, PAMS checks them in lines 18 to 26. If there is a newly arriving data of the same sequence in dataSet, PAMS examines the existence of ItemID and StreamID on the path from root to the current node in line 19. If the ItemID and StreamID is not on the path, PAMS follows the procedure from line 20 to line 26. In line 20, it checks for child node with same ItemID and StreamID. If not found, then PAMS adds a new child with the ItemID, StreamID and timestamp of corresponding sequence of node in line 26. But if found, and if the sequence ID is already in the list, the timestamp is updated as the new timestamp, otherwise new sequence is added in the list. In lines 27 to 28, if the sequence list is empty, this node along with its children can be deleted immediately. But if the node is child of root, the hash entry for this node from the hash table is also removed in lines 29 to 31. Then, if the support of the node satisfies the minimum support, combination of ItemID and streamID of the path from root to this node is listed in output as a frequent sequential pattern across multiple streams in lines 34 to 35.

4 Performance Evaluation

We used IBM synthetic data generator to generate the dataset and transformed for multiple streams. Dataset S5T50I1C1D0.1 represents the dataset with 5 streams, 50 time points, average items per transaction 1, 1000 sequences, and 100 different items. We used C++ to write IAspam and PAMS, and executed the experiments on a computer with core i7 3.4 GHz processor and 12 GB RAM.

The total execution time and memory usage of both algorithms which we executed on S5T50I1C1D0.1 dataset with different minimum support are shown in Fig. 4 whereas with different POI and minimum support 1 % are shown in Fig. 5. Since PAMS generates the same number of candidate sequential patterns

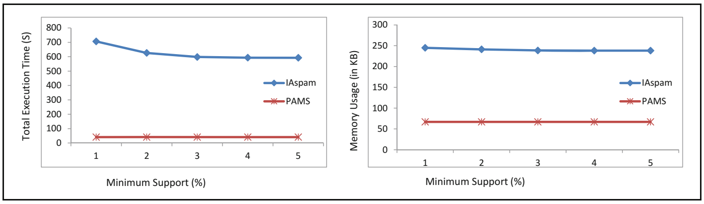


Fig. 4. Total execution time and memory usage with different minimum support and POI 10 on dataset S5T50I1C1D0.1

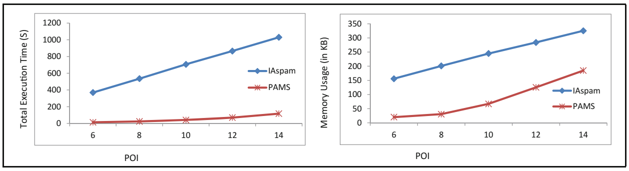


Fig. 5. Total execution time and memory usage with different POI and min. support 1 % on dataset S5T50I1C1D0.1

and PSM-tree irrespective of minimum support, total execution time and memory usage of PAMS remains same in different minimum support. Since IAspam stores bitmap representation and needs to generate more and more candidates in I-step and S-step extension when minimum support decreases, total execution time and memory usage of IAspam increases as minimum support decreases.

5 Conclusion

In multiple data streams, sequential patterns can be found across streams. For mining such sequential patterns in multiple data streams, we have proposed PAMS algorithm. PAMS needs single scan of the database to maintain the PSM-tree for progressive sequential pattern mining across multiple data streams. The experimental results show that PAMS outperforms previous algorithm IAspam.

References

1. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings of the Eleventh International Conference on Data Engineering, pp. 3–14, March 1995
2. Ayres, J., Flannick, J., Gehrke, J., Yiu, T.: Sequential pattern mining using a bitmap representation. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 429–435, July 2002
3. Chen, G., Wu, X., Zhu, X.: Sequential pattern mining in multiple streams. In: Fifth IEEE International Conference on Data Mining, pp. 585–588, November 2005
4. Cheng, H., Yan, X., Han, J.: Incspan: incremental mining of sequential patterns in large database. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 527–532 (2004)
5. Ho, C.C., Li, H.F., Kuo, F.F., Lee, S.Y.: Incremental mining of sequential patterns over a stream sliding window. In: Sixth IEEE International Conference on Data Mining Workshops, pp. 677–681, December 2006
6. Huang, J.W., Tseng, C.Y., Ou, J.C., Chen, M.S.: A general model for sequential pattern mining with a progressive database. IEEE Trans. Knowl. Data Eng. **20**, 1153–1167 (2008)
7. Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C.: Mining sequential patterns by pattern-growth: the prefixspan approach. IEEE Trans. Knowl. Data Eng. **16**(11), 1424–1440 (2004)

8. Wu, W., Gruenwald, L.: Research issues in mining multiple data streams. In: Proceedings of the First International Workshop on Novel Data Stream Pattern Mining Techniques, StreamKDD '10, pp. 56–60 (2010)
9. Xu, C., Chen, Y., Bie, R.: Sequential pattern mining in data streams using the weighted sliding window model. In: 2009 15th International Conference on Parallel and Distributed Systems (ICPADS), pp. 886–890 (2009)
10. Yang, S.Y., Chao, C.M., Chen, P.Z., Sun, C.H.: Incremental mining of across-streams sequential patterns in multiple data streams. J. Comput. **6**, 449–457 (2011)

Efficient Gossiping Dissemination in Mobile Peer-to-Peer Environment Based on Neighborhood Detection

Yu-Jen Lin[1], Bo-Heng Chen[1], Kun-Ta Chuang[1](✉), and Chao-Chun Chen[2]

[1] Department of Computer Science and Information Engineering,
National Cheng Kung University, Tainan, Taiwan
{p76001378,ktchuang}@mail.ncku.edu.tw, bhchen@csie.ncku.edu.tw
[2] Institute of Manufacturing Information and Systems,
National Cheng Kung University, Tainan, Taiwan
chaochun@mail.ncku.edu.tw

Abstract. We in this paper explore a new data dissemination framework in Mobile P2P networks based on neighborhood detection. Previous works in the literature usually elaborated upon the reduction of amount of messages disseminated in the network. However, an important and practical issue remains unresolved. The success of the system design usually relies on the support of other internal sensors such as GPS and accelerator, causing the extra power consumption. In this paper, we propose the NCS framework (standing for Neighborhood Conscious Scheme), which observes user's neighborhood situation for estimating the surrounding environment without using any internal sensor. In our method, NCS not only achieves the low power consumption but also reduces unnecessary redundant messages as compared to the state-of-the-art solution.

1 Introduction

Recently, owing to the rapid advent of mobile wireless communication technologies and the increasing number of mobile users, mobile devices, such as smart phones, wireless Personal Digital Assistants (PDAs), and Tablet PC, have become indispensable in the daily life. On the other hand, the significant advancements in manufacturing low-cost, affordable mobile devices equipped with powerful computing capabilities and a variety of communication radios (e.g. 3G, WiFi, Bluetooth), the way of communications, such as the phone talk and email, has changed from wired to wireless nowadays. It also becomes possible to disseminate of location-aware information in real time via the online media such as social networks or popular location-based services. How to efficiently and effectively deliver various events to users is deemed as the key to the success of the applications.

Recently, the growing market of location-aware advertisements on mobile devices calls for the development of new information sharing medium, called

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 156–167, 2014.
DOI: 10.1007/978-3-319-13186-3_15

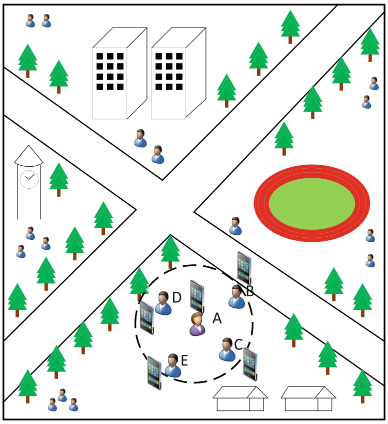


Fig. 1. Application scenario in the Mobile P2P network.

Mobile Peer-to-Peer Networks [6]. The MP2P network is composed of mobile nodes. Specifically, all nodes can make the wireless link with nearby devices in a multicast manner. As such, devices can share and exchange events of interests in time, achieving cost-effective dissemination of instant advertisements in the local area.

An example application scenario shows in Fig. 1. On the campus, user A initiates the data transmission and sends the broadcasting message. Other devices located within its communication range will receive the push request. In this case, users B, C, D and E will receive the message. As time advances, the message will be rebroadcasted by all users who ever received the message. The message will be iteratively rebroadcasted until its expiration. Eventually, the message will be delivered to all users in the network as expected.

Due to the potential profitability of data sharing in Mobile P2P, recent researches elaborated upon the improvement on the effectiveness of data dissemination [2,5,6,8,12–14]. Most previous works focus on how to reduce the amount of data transmissions in the whole network and also keep the high delivery rate (the percentage of mobile users that receive the data successfully). However, the current solution of data dissemination in the mobile P2P network did not consider an important and practical constraint from the hardware support. Previous works usually require the support of additional information, such as the aforementioned location and movement information, which must be acquired by extra hardware components of Global Position System (GPS), Gyroscope, accelerometer sensors, and so on. Despite these internal sensors are available in mobile phones nowadays, enabling these sensors will drain the battery of the phone in a few hours. It inevitably increases the power consumption [6,12]. Such results are undesired but are generally ignored in previous works.

We consider a framework to achieve energy-saving dissemination in mobile peer to peer networks. The proposed framework, called the NCS framework (abbreviated from Neighborhood Conscious Scheme), is designed without the need of supports from extra hardware components. NCS can not only reduce the total amount of transmissions in the network, but also guarantee the high delivery rate. The contributions of our approach include: (1) Since the battery of cell phones will be ran out within a few hours when users turn on the internal sensors (GPS, Gyroscope, accelerometer sensors and so on) [15], we are the first work attempting to achieve effective data dissemination without the need of enabling extra sensors. (2) In NCS, we devise an effective method to avoid unnecessary dissemination by observing user's neighborhood situation. (3) While previous works usually assume the MANET configuration must be accomplished in the broadcast manner, the multicast way is applied in NCS instead, enabling data dissemination both in the Bluetooth or the WiFi environment.

The rest of this paper is organized as follows. Section 2 gives preliminaries including related works. In Sect. 3, we describe the design of our method. The Sect. 4 shows experimental results. Finally, we give conclusions and future works in Sect. 5.

2 Preliminaries

2.1 Related Works

In recent year, many efforts have been devoted to improve efficiency of data dissemination in the Mobile Peer-to-Peer network. In [6], the authors proposed a method aiming at utilizing distance, velocity, and moving direction information for reducing redundant advertising messages in the wireless environment. However, in order to precisely estimate the re-transmission at the right moment, it needs information provided from extra hardware sensors (e.g. GPS, Accelerometer). Clearly, the solution incurs the penalty of considerable power consumption. In addition, another problem about fairness arises in this model. Some users will need to re-transmit messages more frequently as compared to others, resulting in the impracticable concern.

Furthermore, the work of Opportunistic Resource Exchange [14] focuses on disseminating real-time location specific information in inter-vehicle Ad-hoc networks. The spatial and temporal criteria are taken into consideration. These two characteristics help the vehicle to decide whether the data should be stored in a cache and/or broadcasted. Same as the work in [6], the solution in [14] also requires the spatial information about mobile devices, which inevitably needs some sensors such as GPS being turned on. Note that these models assume that the MANET configuration is realized in 3G or WiFi due to the need of broadcasting. Such a requirement of accessing 3G or WiFi everywhere will limit the practicability of the proposed model.

In [12], a Rank-Based Broadcast (RBB) method was proposed, in which a mobile peer dynamically adjusts the P2P transmission size depending on channel utilization and bandwidth utilization. The objective is to maximize the throughput. A transmission is triggered when users received enough new queries or there are a sufficient number of reports. And the ranking strategy (i.e., priority of

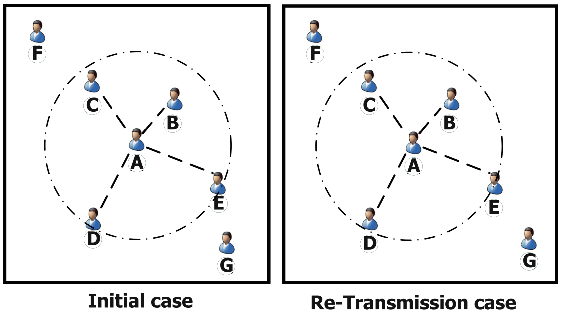


Fig. 2. The scenario that users are stable.

broadcast) is determined for user's queries and information of data. The popular report is most frequently requested by users, and so that the most popular report is always of high priority. In such a way that the user will always broadcast it to other nodes. However, some reports which are not so popular will be disseminated rarely, incurring the low delivery rate of such messages.

In [9], a temporal-spatial data ranking strategy is proposed to determine whether the data should be stored in the database and transmitted to other users who encounter in the future. This method aims at reducing the possibility that users get the identical data. The issue addressed and resolved in [9] merely focuses on the memory and bandwidth/energy constraints. It is related but orthogonal to our work.

2.2 Environmental Description

In this section, we describe the various scenarios of the environment that will happen in our model. Note that in related figures, the square symbol is denoted as the previous position and the circle symbol is denoted as the current position.

Figure 2 shows that there are many users within the source node A's transmission range and the behavior of user movement is relatively slow. Since neighbors of node A may not change awhile, its neighbor will have a higher opportunity to occur again in the next gossip round. The environment is called "stable". In this environment, user A should not retransmit message frequently, because its neighbors will always receive the same messages.

Figure 3 shows the scenario that the source node A moves quickly, and the behavior of other devices movement is relatively slow. Note that in cases of Fig. 3, information is not easy to be disseminated effectively. Many re-transmissions may incur unnecessary transmissions (e.g. user B, C, D, E, and F). In this environment, user A should retransmit message frequently than other users, because user A will meet new neighbors easily in the future.

Figure 4 shows that all devices move quickly. It can be efficiently disseminated outside and it is easy to achieve high delivery rate in such cases.

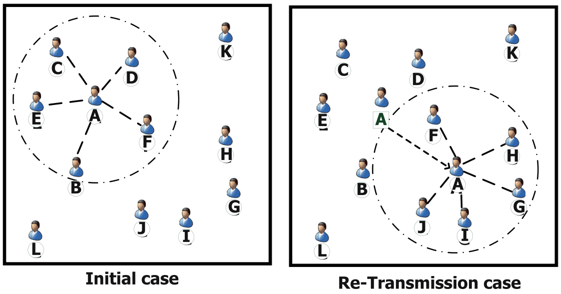


Fig. 3. The source node A moves quickly but the rest of nodes are relatively stable.

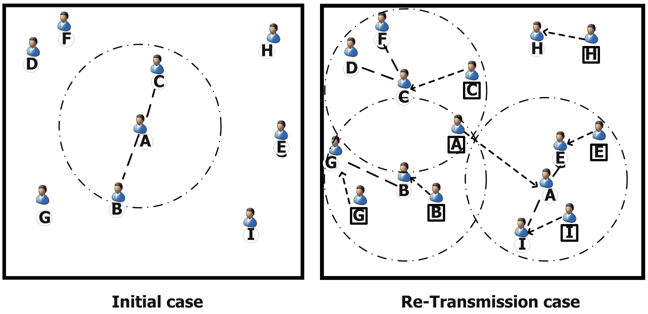


Fig. 4. Cases of the quick movement in the environment.

Finally, Fig. 5 shows that users A, B, C, D, and E have received the same data. During the next gossiping round, user A will receive many duplicate data from B, C, D and E. While user gets many duplicate messages during a period of time, it means that most people already have the same information in this region, we could defer the next gossiping time.

We also give some essential definitions below, and meanings of notations used in this paper are also summarized in Table 1.

Definition 1 (Message): A message m_i is a string content associated with a timestamp, denoted by $M_t(\mathrm{m}_i)$. And the age of the message m_i is denoted by $\mathrm{ttl}(\mathrm{m}_i)$.

Definition 2 (New Arrival Neighbors): Let $\Delta N_t^{t+1}(o_k)$ denote the set of object o_k's new neighbors who arrive in o_k's transmission range during t and t+1 moment. And the number of $\Delta N_t^{t+1}(o_k)$ is denoted by $\left|\Delta N_t^{t+1}(o_k)\right|$.

3 The NCS Framework

In this paper, we propose the NCS framework (abbreviated from Neighborhood Conscious Scheme). We modified Opportunistic Gossiping scheme [3,7,11] to disseminate the instant advertisements. Our method is a neighborhood conscious solution for data dissemination in mobile peer-to-peer environment. We will not

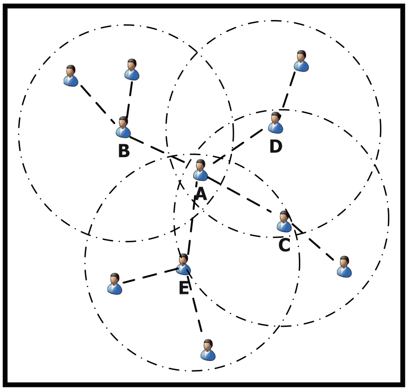


Fig. 5. All users have received data.

Table 1. A list of notations.

Notation	Description
m_i	The i^{th} content distinct message in the system
$Mt(\mathrm{m}_i)$	The timestamp of message m_i
$\mathrm{ttl}(\mathrm{m}_i)$	Time to live of m_i
o_k	The k^{th} moving object in the system
O	$\mathrm{O} = \{\mathrm{o}_1,\mathrm{o}_2,..\mathrm{o}_h\}$
$\mathrm{N}_t(\mathrm{o}_k)$	The set of neighbors of o_k in time t
$\lvert N_t(o_k)\rvert$	The number of neighbors of o_k in time $_t$, i.e., the density value
$\varDelta N_t^{t+1}(o_k)$	$\varDelta N_t^{t+1}(o_k) = N_{t+1}(o_k) - N_t(o_k)$
$\lvert\varDelta N_t^{t+1}(o_k)\rvert$	The number of new arrival users between t and $t+1$ for o_k
$dupl_t^k(\mathrm{m}_i)$	The number of receiving duplicate messages m_i in object ok at time t
$\varDelta t$	Gossiping round time

require the support from any internal sensors in phones. In order to detect the environment which we describe in Sect. 2.2, we will record the information about its neighbors. Once a user estimates the possible scenario of the environment which it may belong to, it will automatically adjust the broadcast period to reduce redundant messages. Figure 6 below depicts our system flowchart.

As shown in Fig. 6, when users receive the message, it will record the information about the object which sent this message and the times of receiving this message. The objective of this step is for reducing redundant messages, therefore we must avoid unnecessary transmissions. Users will send the message until the next gossiping time.

The previous work such as the Optimized Gossiping approach has encountered some challenges. We take the following figure as the example. People tend to move in group today (e.g. student walking on the campus or people shopping

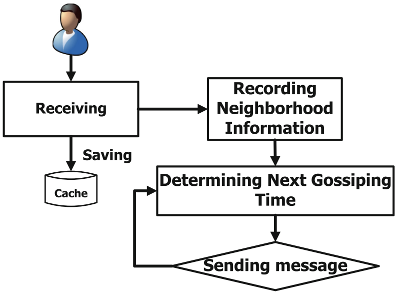

Receiving
Recording Neighborhood Information
Saving
Cache
Determining Next Gossiping Time
Sending message

Fig. 6. Flowchart of NCS.

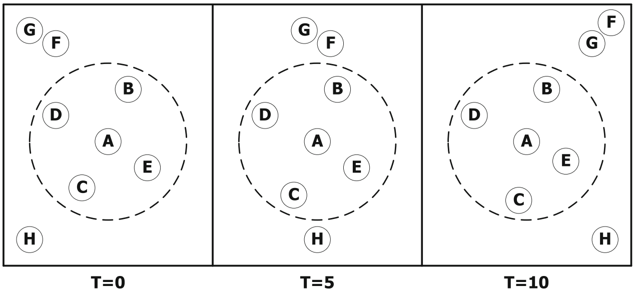


Fig. 7. The scenario of the unfair power consumption issue.

in malls). As shown in Fig. 7, while user A broadcasts a message to neighbors, at T = 1, users B, C, D and E receive the new message. When T = 5, user A broadcasts the same message again, using the traditional method, it will postpone the scheduled time for the next message transmission. Because B, C, D and E receive the duplicate data, they should delay for a while to avoid unnecessary transmissions. At T = 10, they act similar as that at T = 5. Therefore, user A will broadcast more frequently than others, but users B, C, D and E will do nothing and do not take the appropriate responsibility for message dissemination. In addition, in this scenario, users F, G, and H will not get new data since user B will continuously postpone the scheduled gossiping time. It is not a desired property in data dissemination in the P2P environment.

As mentioned above, there are some side-effects of these approaches. First, it may cause the fairness issue in this scenario, some users (e.g. user A) will always disseminate messages to neighbors, but other users (e.g. users B, C, D and E) postpone the scheduled gossiping time. Second, while some users pace near the group (e.g. users H and G), they may lose chance to get new messages.

In order to tackle these issues, our method will observe user's neighborhood variation to determine the scheduled time for next gossiping. The user will postpone scheduled time for a short period when he/she finds that his/her neighbors have changed dramatically. In such a way that user can propagate new messages to new neighbor quickly. Conversely, while his/her neighbors have not changed or merely changed slightly, users will postpone scheduled time for a long period. It can avoid unnecessary transmission and also reduce redundant messages. Therefore, we develop formula to describe how to determine the next gossiping time T$_{next}$. Here T$_{now}$ is the current scheduled time for gossiping, and R_{na} denotes the percentage of number of new arrival neighbors to total number of neighbors. R_d denotes the count of duplicate messages received by a user during a certain period time.

$$T_{next} = \begin{cases} T_{now} + \Delta t \times (\frac{1+R_d}{R_{na}}), \, if \; R_{na} > 0 \\ \\ T_{now} + \Delta t \times (\frac{1+R_d}{0.1}), \, otherwise \end{cases},$$

where

$$R_{na} = \frac{\left|\Delta N_t^{t+1}(o_k)\right|}{N_{t+1}(o_k)},$$

and

$$R_d = \frac{dupl_t^k(m_i)}{(T_{now} - T_{before})}.$$

As a result, if there are few new neighbors come into the transmission range, the value of R_{na} will be smaller. We estimate that the user may locate in the stable environment. So that he/she will not easily meet the new neighbors during the upcoming period. The user must wait for a longer period of time and retransmit the message. Thus, it reduces the probability that the user's neighbors receive the same messages. It is helpful for us to reduce redundant messages in the system. Note that when the user receives duplicate messages frequently, the value of R_d will be high. It means that most users have received this message in this region. In such cases, user should not disseminate this message frequently.

To sum up, because of our method will record the neighbors information to determine the scheduled time for next gossiping. In this way, while users with an unstable environment, he/she will retransmit the information to his/her neighbors at the right time. It can help user to reduce the number of times of unnecessary retransmission and also save energy in the same time. In opposition to the Optimized Gossiping approach, they will encounter the unfairness problem. When the users in certain region, he/she will always need to retransmit data to his/her neighbors. Some of user must be consuming more energy than others. And the retransmission may not bring any benefits to his/her neighbors, because the neighbors will not receive any new information.

4 Experimental Results

In this section, we firstly analyze and compare the performance of our proposed NCS algorithm and Optimized Gossiping approach and Opportunistic Gossiping [6]. Then we further discuss our model performance in different parameters.

These three approaches, i.e., Opportunistic Gossiping, Optimized Gossiping and Neighborhood Conscious Scheme, are all implemented as the broadcasting protocol in Network simulator NS-3, which is a discrete event network simulator and is completely designed in the C++ programming language. Note that NS-3 is one of the most widely used simulator in networking research area [1,9,10]. We evaluate the performance of these methods in a region of 3000 m × 3000 m mobile scenario according to the Random Waypoint mobility model [4]. This model is also the most commonly used mobility model in networking research area. In this model, each moving node uniformly distributed of the initial state in the simulation area. And each node is moving at a constant speed along a straight line from its current position to its randomly selected destination. When a mobile node reaches his destination, it pauses for a fixed amount of time and then moves again to a new random destination. All mobile users communicate with the 802.11 protocol in the multicast manner. The values of parameter settings are listed in Table II, and there are three metrics to measure the performance of our approach:

- Delivery Rate: The percentage of mobile users that receive the message successfully when passing through the advertising area. The ideal value of Delivery rate should be close to 100 %. The value of delivery is determined as

  $$Delivery\,rate = \frac{U_g}{U_p},$$

  where U_g denotes number of users got the message and $U_p$the number of user passed through the advertising area.
- Delivery Time: It is estimated by the time that users receives the message. The shorter delivery time means that the message is delivered more efficiently to new users. Note that the shorter delivery time leads to the better performance.
- Number of messages: It denotes the total number of messages generated by the all users in the advertising area. Reducing the number of messages will not only save wireless bandwidth but also relieve network congestions. The number of messages should be reduced as much as possible.

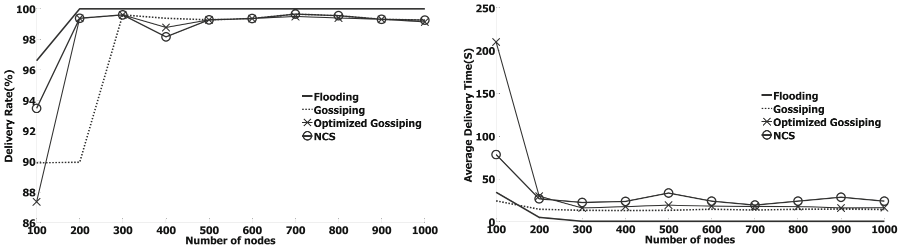
4.1 Performance Comparison

In this section, we discuss the performance of our approach in different network sizes and different transmission ranges. We issue only one message at the center (1500, 1500) of the simulation area.

We first compare Delivery Rate of Flooding, Gossiping, Optimized Gossiping and NCS with different network sizes (100 peers to 1000 peers). And the transmission range is 250 m. As illustrated in Fig. 8(a), the delivery rate is high no matter the network is sparse or dense. Because all users behave with the wide range of communication (250 m), it is easy to cover the whole advertising area as time advances. While users pass through the advertising area, he/she will easily receive the data. Hence, the delivery rate of these approaches could keep good delivery rate (Table 2).

Table 2. Parameter setting.

Parameter name	Value
Simulation time	2000 sec
Number of peers	100,200,300,400,500,600,700,800,900,1000
Radius of advertising area	100 m
Duration of message	1800 sec
Gossiping round time	5 sec
Transmission range	250 m (802.11)
Number of multicast members	7



(a) Delivery Rate

(b) Average Delivery Time

Fig. 8. Delivery rate and average delivery time in different network sizes.

As presented in Fig. 8(b), Flooding outperforms the other approaches in average delivery time in the sparse network. It is because messages must be immediately retransmitted in the Flooding algorithm when user receives a message. It will disseminate information quickly in the whole network. In Gossiping, users will disseminate data at a fixed period. Users also need to transmit frequently in the advertising area. When the network size is larger than 200 peers, the performance results of these approaches are similar.

As shown in Fig. 9, we can see that number of messages generated by NCS is only 17 % and 79 % of that generated by Gossiping and Optimized Gossiping when the network contains 1000 peers. Obviously, our approach significantly improves the performance of reducing messages as compared to Flooding and Gossiping. And NCS also generates the fewer number of messages than the Optimized Gossiping.

As illustrated in Fig. 10, each red column represents the percentage of messages reduced comparing with Gossiping. It is clear that NCS apparently outperforms the Gossiping algorithm. And each blue column represents the percentage of messages reduced from Optimized Gossiping. Except when the network size is 100, the percentage of the amount of messages is reduced more than 10–30 % in general when the NCS algorithm is applied, thus showing its high practicability.

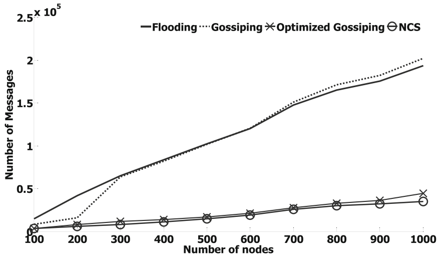


Fig. 9. Number of messages in different network sizes.

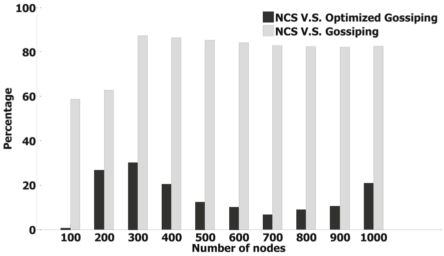


Fig. 10. Percentage of messages reduced by NCS.

5 Conclusions

To conclude, the proposed work is a new research direction on how to disseminate data efficiently without the hardware support from internal sensors (e.g. GPS, Gyroscope and accelerometer) in the MP2P network. By detecting the scenario of the user environment, all of the advertisements would be disseminated in a simple way. Our experimental results show that the proposed algorithm, called NCS, provides the high delivery rate of messages while keeping the low delivery time and the low count of messages transmitted in the system. In addition, for the sparse and the dense network, NCS can generate a lower number of advertisements than Flooding and Gossiping. In the un-sparse network, NCS also achieves the high delivery rate as good as other state-of-the-art algorithms in the literature while also reducing the number of messages as possible. It is worth mentioning that the energy consumption of mobile devices can be reduced obviously in the NCS algorithm.

The future enhancement should be focused on the update issue and multi-source applications. Such a scenario usually appears in the real world, and the advertisements are generated by any user with updated content, e.g., Soccer scores or the remaining number of parking lots. Current solutions all assume the static content, and cannot easily to be extended to such applications.

Acknowledgement. This paper was supported in part by National Science Council of Taiwan under Contract NSC 101-2221-E-006- 246-MY3, and the Ministry of Economic Affairs under the grant 101-EC-17-A-02-S1-222.

References

1. ns-3 tutorial (2009). http://www.nsnam.org/
2. Balasubramanian, N., Balasubramanian, A., Venkataramani, A.: Energy consumption in mobile phones: a measurement study and implications for network applications. In: IMC, pp. 280–293 (2009)
3. Boyd, S., Ghosh, A., Prabhakar, B., Shah, D.: Gossip algorithms: design, analysis and applications. In: INFOCOM, pp. 1653–1664. IEEE (2005)
4. Camp, T., Boleng, J., Davies, V.: A survey of mobility models for ad hoc network research. Wirel. Commun. Mob. Comput. **2**(5), 483–502 (2002)
5. Chen, L., Cui, B., Shen, H.T., Lu, W., Zhou, X.: Efficient information retrieval in mobile peer-to-peer networks. In: CIKM, pp. 967–976 (2009)
6. Chen, Z., Shen, H.T., Xu, Q., Zhou, X.: Instant advertising in mobile peer-to-peer networks. In: ICDE, pp. 736–747. IEEE (2009)
7. Conti, M., Kumar, M.: Opportunities in opportunistic computing. IEEE Comput. **43**(1), 42–50 (2010)
8. Henderson, T.R., Lacage, M., Riley, G.F.: Network simulations with the ns-3 simulator. In: SIGCOMM'08 Demonstration (2008)
9. Luo, Y., Wolfson, O., Xu, B.: A spatio-temporal approach to selective data dissemination in mobile peer-to-peer networks. In: ICWMC, p. 50b (2007)
10. Naicken, S., Livingston, B., Basu, A., Rodhetbhai, S., Wakeman, I., Chalmers, D.: The state of peer-to-peer simulators and simulations. Comput. Commun. Rev. **37**(2), 95–98 (2007)
11. Pelusi, L., Passarella, A., Conti, M.: Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. IEEE Commun. Mag. **44**(11), 134–141 (2006)
12. Wolfson, O., Xu, B., Yin, H., Cao, H.: Search-and-discover in mobile p2p network databases. In: ICDCS, p. 65. IEEE (2006)
13. Wolfson, O., Xu, B., Yin, H., Cao, H.: Searching local information in mobile databases. In: Liu, L., Reuter, A., Whang, K.-Y., Zhang, J. (eds.) ICDE, p. 136. IEEE (2006)
14. Xu, B., Ouksel, A., Wolfson, O.: Opportunistic resource exchange in inter-vehicle ad-hoc networks. In: Mobile Data Management, pp. 4–12. IEEE Computer Society (2004)
15. Youssef, M., Yosef, M.A., El-Derini, M.: GAC: energy-efficient hybrid GPS-accelerometer-compass GSM localization. In: GLOBECOM, pp. 1–5. IEEE (2010)

Predicting Locations of Mobile Users Based on Behavior Semantic Mining

Huei-Yu Lung, Chih-Heng Chung, and Bi-Ru Dai(✉)

National Taiwan University of Science and Technology, #43, Sec.4, Keelung Rd,
Taipei 106, Taiwan, Republic of China
jeremyfederer0608@gmail.com,
D9915015@mail.ntust.edu.tw, brdai@csie.ntust.edu.tw

Abstract. Predicting movements of mobile users has become increasingly popular because of the ease of trajectory data collecting nowadays. However, as most of these prediction techniques need geographic pattern matching of users' trajectory data, it is possible that the techniques cannot work in a place where the user has never been before. In this paper, we propose an approach based on transportation mode and behavior semantic features to predict the next location of the users' movement. First, we identify the users' transportation mode to get sequential data of the users' motion mode. Then, we get the semantic meaning as behavior semantic features from the places where users have stopped and visited for a while. We determine the relationship between the transportation mode and behavior semantic features to predict the next location based on the Hidden Markov model. We use real world data for our experiment to demonstrate the effectiveness of our approach.

Keywords: GPS · Data mining · Transportation mode · Behavior semantic labels · Hidden Markov model · Movement prediction

1 Introduction

When mobile networks and mobile phones are combined with the ever-increasing availability of location-acquisition technologies, we have better access to collections of large spatio-temporal datasets. Personal services have become popular and one rapidly growing example in recent years is location based service (LBS). To provide precise service, it is important for LBS to be able to predict the activities that the users want to do at the next location. For this reason, effective activity and location prediction techniques for LBS are necessary.

Many studies use pattern-based prediction methods to predict the next location from the mobile users' GPS trajectories, but these prediction methods usually require the anticipated movement to be a full match with the pattern to make the prediction. Other studies analyze the frequent patterns of mobile users, but most of these only consider the geographic features [16]. When the prediction is only based on the geographic trajectory, consisting of a sequence of geographic points and timestamps, it is possible to be misled by the geographic distance and trajectory shape. For example, as Fig. 1 shows, some prediction techniques would predict the destination of trajectory1

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 168–180, 2014.
DOI: 10.1007/978-3-319-13186-3_16

based on its geographical similarity to trajectory2. Yet, as we can see from Fig. 1, the next movement of trajectory1 is more like the destination of trajectory3. A solution to this problem involves the notion of semantic trajectory [4, 5], which consists of a sequence of locations labeled with semantic tags, from which we can infer the users' activities. From Fig. 1, we can tag trajectory1 as <school, bank, hospital>. Although previous research has used semantic labels for supporting, geographic feature data are still required to be matched with the history training data.

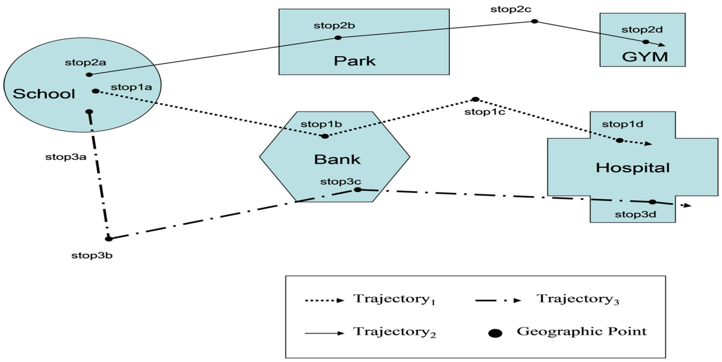


Fig. 1. An example of semantic trajectory.

In this paper, we propose a novel location prediction framework that makes three major contributions: (i) we propose a location prediction strategy to predict the next location of the users based on the users' transportation mode and behavior semantic labels; (ii) we use the Hidden Markov Model to find the relationship between the users' transportation mode and behavior semantic labels; (iii) we do not require history trip data for geographic mining. We can make a prediction even if the mobile users have never been to this place before and no other training data exist.

The remainder of this paper is organized as follows. We discuss several prediction strategies and some algorithms related to our work in Sect. 2. In Sect. 3, we introduce the proposed approach called TransemanPredict, for which Experimental evaluations are presented in Sect. 4. Finally, conclusions are made in Sect. 5.

2 Related Works

In this section, we will introduce the related works to our prediction method.

2.1 Movement Prediction

In order to predict the next location of mobile users, data mining techniques are the appropriate method to be employed. The personal-based prediction [17] and the general-based prediction [3–5, 16, 25] are often adopted in this problem.

The personal-based prediction approach uses only the movements of an individual user to anticipate the next location because it considers the movement of each individual user as independent. In [17] proposes the Individual Life Pattern which models the users' periodic behaviors by mining from the individual trajectory data.

On the other hand, the general-based approach makes a prediction based on every movement behavior of the general mobile users. In [3] proposes a method to extract the user's behavior pattern by classifying frequent-stay locations and moving locations and by using precalculated Destination Reference Data to determine the next destination. In [4] suggests a novel approach based on both the geographic and semantic features of the user's trajectory to predict the next location. They use a cluster-based prediction strategy to evaluate the destination based on the frequent behaviors of similar users in the same cluster. In [5] presents approaches to discovering personal mobility and characteristics based on location and semantic information. In [16] introduces a method to predict the next location of a moving object with a certain level of the accuracy. In [25], both the intended destination and the future route of a person are predicted.

In our approach, we choose the general-based prediction techniques.

2.2 Transportation Mode Detection

Transportation mode detection techniques have been developed to recognize and understand human behavior in recent years [9, 10, 15]. In [9] proposes an approach based on supervised learning to infer people's motion modes from their GPS logs. In [10] introduces an algorithm based on real world observations and knowledge extracted from collected GPS sensor data to automatically identify mobility transfer points. In [15] puts forward an approach that constructs the users' mobility profiles and calculates the mobility similarities between users.

2.3 Hidden Markov Model

The Hidden Markov Model (HMM) is a powerful statistical tool for modeling generative sequences that can be characterized by an underlying process generating an observable sequence [18, 19]. In this way, it has already been used to predict the future location of mobile users [6–8]. In [6] proposes a novel approach named Trajectory Pattern Models to explain the relationship between the frequent regions and the partitioned cells in order to predict the next location. In [7] presents a hybrid method for predicting human mobility on the basis of HMMs. In [8] puts forward an application of the shared structure Hierarchical Hidden Markov Model (HHMM), which is the simultaneous estimation of the model's parameters at all levels, and a construction of a Rao-Blackwellised particle filter (RBPF) approximate inference scheme. However, insofar as these prediction techniques do not consider the semantic features to make the prediction, it is possible that those techniques cannot work for the places where users have never been before.

3 Location Prediction Model

To support the LBS of mobile user, we propose a novel location prediction framework named TransemanPredict, based on the mobile users' transportation mode and behavior semantic features extracted from trajectories. We want to make the prediction based solely on the user's mobility behaviors. In this section, we will introduce our approach in five parts. The architecture of our framework can be seen in Fig. 2.

The first step, called Data Preprocessing, extracts the user's stay location and transfers the trajectory into Stay Location Sequences. The second step, called Transportation Mode Detection, identifies the mobile user's transportation mode. We tag the moving segment trajectory into Transportation Mode Sequence, e.g., <walking, bus, MRT, walking>. The third step, called Behavior Semantic Label Tagging, uses the Google map API to search for the most popular location near the staying location of the user as its behavior semantic label, generates the user's own semantic label such as Home, Work office, etc., and then transforms the Stay Location Sequences into a Behavior Semantic Label Sequence, e.g., <home, school, restaurant, park>. In the fourth step, called Semantic Mining Model, the relationship between the transportation mode and behavior semantic features is found in order to predict the next location based on the Hidden Markov Model. After the fourth step, the user's next most probable behavior semantic location is obtained. To predict the real trajectory point, we use the Google API in average transportation distance to find several probable semantic trajectory locations. Then we use the Heading Change Rate and the user's average travel time in certain transportation mode situations to evaluate the most probable location as our prediction.

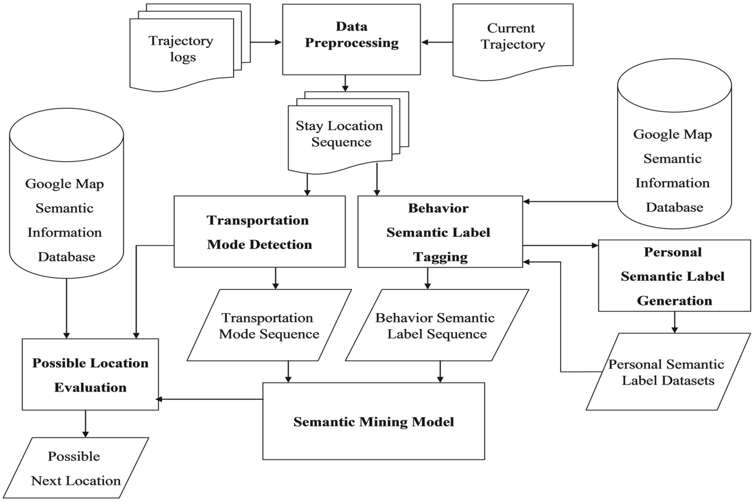

Trajectory logs
Data Preprocessing
Current Trajectory
Stay Location Sequence
Google Map Semantic Information Database
Google Map Semantic Information Database
Transportation Mode Detection
Behavior Semantic Label Tagging
Personal Semantic Label Generation
Transportation Mode Sequence
Behavior Semantic Label Sequence
Personal Semantic Label Datasets
Possible Location Evaluation
Semantic Mining Model
Possible Next Location

Fig. 2. The TransemanPredict framework for location prediction.

3.1 Data Preprocessing

Given that users usually do activities depending on where they are, we first have to transfer the trajectory logs into a stay location sequence. By doing this, we can separate the moving part and the staying part of the trajectories. The benefit of doing this is that we can focus on the segment of the trajectory that we want to handle while simultaneously decreasing the amount of our data. Since we have to define the stay location, we set a time threshold and a distance threshold. The time threshold defines a place with a minimum of time stayed, whereas the distance threshold defines the maximum distance the user stayed over the time threshold. As such, if the user stays in a place where the distance between a trajectory log and another trajectory log is the distance threshold and for a duration which is over the time threshold, we define this place as a stay location. We follow [21] to discover stay points from the user's GPS trajectories and set 30 min as the time threshold and 200 m as the distance threshold.

3.2 Transportation Mode Detection

To predict the next location of the users, understanding the users' behavior is essential. As a kind of user behavior, the transportation modes, such as walking, driving, etc., can enrich the user's mobility with informative knowledge and provide pervasive computing systems with more context information. Since transportation mode is one kind of human behavior, [9] proposes several sophisticated features based on human experience. The features are listed as follows:

1. Dist: Distance of a segment.
2. V: Velocity of trajectory log.
3. A: Acceleration of trajectory log.
4. MaxV: The maximum velocity of segment.
5. MaxA: The maximum acceleration of segment.
6. AV: Average velocity of a segment.
7. SR: Stop Rate. In our experience and observation, people in different transportation modes will stop at different times. For example, people who take the bus will stop more times than those who drive because the bus needs to stop for passengers at bus stops. We set a velocity threshold V_s for detecting the stop points.
8. VCR: Velocity Change Rate. We set V_r as the velocity change threshold.
9. H: Heading Change.
10. HCR: Heading Change Rate. Regardless of whether they are by car or bus, these transportation modes are still constrained by the road. People cannot change the direction in which they are heading as easily as if they were walking or riding a bicycle. These observations thus motivated us to use HCR to identify the transportation modes. We define a complete heading change as the collection of GPS points from which users change their heading direction exceeding a certain threshold (H_c).
11. TM: Transfer Mode

We use above features to identify the transportation modes. The processing steps are shown in Fig. 3 and explained below:

Step 1: We use a threshold of velocity (V_w) and acceleration (A_w) to identify the non-Walk segment. If a segment contains a velocity or acceleration beyond what a human is capable of, it is obvious that this segment is definitely a non-Walk segment.

Step 2: In our datasets of GEOLIFE [1], we have training data with each segment already being tagged by people. For this reason, we use several classification algorithms such as the Naïve Bayes classifier [26], J48 classification algorithm [24] based on a decision tree and Support Vector Machine (SVM) [27]. Training our model based on supervised learning, the threshold we set is after [9]. Here, H_c is set to 19 degrees for HCR, V_s is set to 3.4 m/s for SR, and V_r is set to 0.26 for VCR.

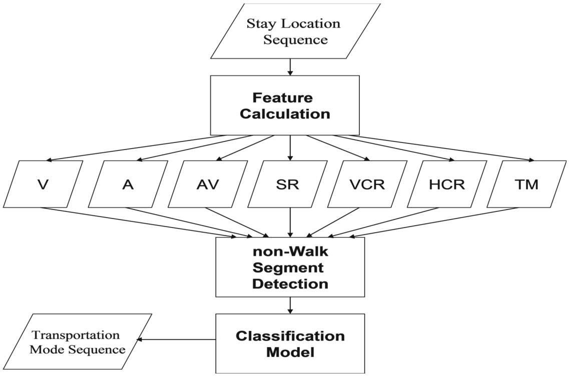


Fig. 3. The framework of transportation mode detection.

We choose the J48 classification algorithm to classify the transportation mode because each feature we chose has the capability of classifying a transportation mode. We still use Naïve Bayes classifier and SVM as comparisons for verification.

There is a connection between the transportation mode and the destination because the same type of places usually has more or less the same traffic conditions and geographical locations. For example, people usually will not walk or take a bus to a shopping mall because they probably need to carry many things when they leave the mall. As such, we select this feature to make our strategy more precise in our approach, which stands in contrast to all of the other approaches that ignore it.

3.3 Behavior Semantic Label Tagging

Some research suggests approaches to the understanding or discovering of human semantic places [11, 14]. However, in our approach, we use a service of the Google Maps API [22], called Nearby Search Requests, to tag the semantic label of the stay point location. We can refine our search request by supplying keywords or specifying the type of places we want to find. There are some of the possible search parameters:

- **Location:** latitude/longitude. Enter the GPS log of the stay point location.
- **Radius:** distTh. We set *disThres* (200 m) as the radius.
- **Rankby:** Prominence, Distance. We choose Prominence, which means sorting results based on their importance. Prominence can be affected by the ranking of the place in Google's index, the number of check-ins from our application, global popularity or other factors.

- **Types:** Restricts the results to place matching at least one of the specified types. We select 10 types of places as our semantic labels: school, restaurant, movie theater, hospital, park, department store, bank, museum, library and gym.

We can use the history of the user's trajectory datasets to generate the user's own personal semantic labels. We follow Xie et al.'s approach to find out the user's Region of Interests (ROI) and then apply the ROIs to the user, considering the time of day, the day of the week, and public holidays, to generate personal semantic label such as Home, Work office. Semantic labels have been used to predict the next location [4], but unknown labels are sometimes produced, which is not helpful.

3.4 Semantic Mining Model

In this model, we try to mine the relationship between the transportation mode and behavior semantic label. We also follow [24] to mine behavior semantic patterns by using transportation mode and behavior semantic label.

We chose Hidden Markov Model (HMM), a well-known approach for the analysis of sequential data, to predict the next location because there are meaningful transition probabilities between each behavior semantic label and transportation mode. In order to construct a trajectory pattern model like that depicted in Fig. 4, we must define the elements of an HMM, which are (i) a hidden state, (ii) observable symbols, (iii) an initial state, (iv) a set of state transition probabilities, and (v) output probabilities.

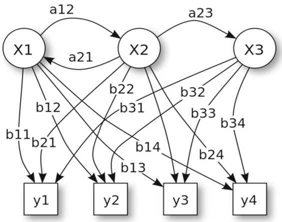


Fig. 4. Probabilistic parameter of hidden Markov model. x - hidden state, y - observable symbols, a - state transition probabilities, b - output probabilities.

We define behavior semantic labels as hidden state S = {s_i, s_i, s_i, …, s_i}. And then we define the transportation mode before each stayed location as observable symbols sequence O = {o_i, o_i, o_i, …, o_i}. The state transition probabilities can be expressed as a matrix A = {a_{ij}}, where $a_{ij} = P[s_{t+1} = S_j \mid s_t = S_i]$, $1 \le i, j \le$ (Number of Semantic Label types). The output observation transition probabilities can be expressed as a matrix B = {b_{ij}}, where $b_{ij} = P[o_t \mid s_t = S_i]$, $1 \le i \le$ (Number of state), $1 \le j \le$ (Number of Transportation Modes). The initial state is defined as $\pi = \{\pi_i\}$ $1 \le i \le$ (Number of Semantic Label types). By HMM, we can get a possible behavior semantic label as the possible movement location of the mobile user.

3.5 Possible Location Evaluation

Since we get a behavior semantic label as our prediction result, we need to change it into a real GPS trajectory log to find the next destination of the mobile user. The behavior semantic label has the same semantic meaning as using the Google Maps API to search for a location, with the search radius being dependent on the average transfer time based on the transportation mode. After the search, some candidate locations are created and evaluated by the heading change rate. We choose the best candidate locations as our prediction. Figure 5 illustrates this framework.

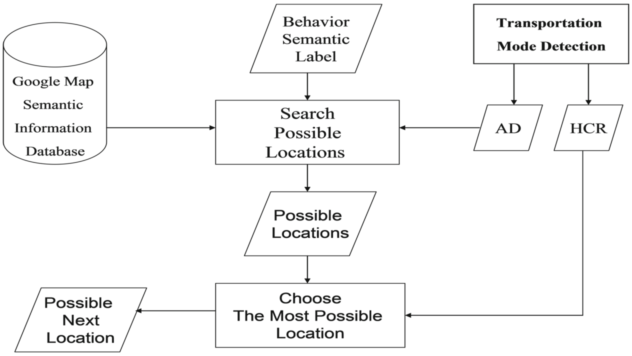

Google Map Semantic Information Database
Behavior Semantic Label
Transportation Mode Detection
Search Possible Locations
AD
HCR
Possible Locations
Possible Next Location
Choose The Most Possible Location

Fig. 5. The framework of possible location evaluation.

4 Experiment Study

In this section, we describe the experiment results of our prediction technique using the GEOLIFE dataset.

4.1 Datasets

We use the Microsoft Research Asia Geolife project [1] datasets version 1.3. Sixty-nine users' trajectory datasets tagged with transportation mode labels are used, covering a total length of 140304 km and a total duration of 12953 h.

4.2 Transportation Mode Detection

We use 5-fold cross validation to verify each classification model. Table 1 presents the results of the J48 classification algorithm, Table 2 the results of the Naïve Bayes classification algorithm, and Table 3 the results of the SVM classification algorithm. The accuracy of the approach of [9] is 0.762.

In Tables 1 and 3, we can observe that the accuracy of the classification of the bus mode is the highest. This is because the bus has the most regular transportation pattern and the largest amount of data. Our method seems confused on classifying the mode of

Table 1. Confusion matrix of transportation mode detection.

		Actual class				
		Walk	Bike	Car	Bus	
Predicted class	Walk	7104	2318	1	23	
	Bike	1785	5320	23	783	
	Car	3	24	631	1632	
	Bus	67	412	146	17643	
	Accuracy	0.793	0.659	0.788	0.878	0.779

Table 2. Confusion matrix of transportation mode detection by using Naïve Bayes.

		Actual class				
		Walk	Bike	Car	Bus	
Predicted class	Walk	6307	2359	13	228	
	Bike	2148	5229	21	1475	
	Car	24	56	649	3874	
	Bus	480	430	118	14504	
	Accuracy	0.704	0.648	0.810	0.722	0.721

Table 3. Confusion matrix of transportation mode detection by using SVM.

		Actual class				
		Walk	Bike	Car	Bus	
Predicted class	Walk	7267	2233	2	169	
	Bike	1643	5186	14	632	
	Car	19	121	646	1445	
	Bus	30	534	139	17835	
	Accuracy	0.811	0.642	0.806	0.888	0.787

walking and biking, which is because of their high similarities on SR, VCR and HCR. The car has higher velocity, and thus is seldom classified as either walking or biking modes. In Table 2, we can observe that the accuracy of transportation mode detection by using Naïve Bayes is the lowest.

4.3 Comparison of Prediction Strategies

Here is the precision analysis between SemanPredict [4] and the use of the semantic mining strategy of SemanPredict on features as selected from transportation mode labels, behavior semantic labels and time semantic labels (where the time semantic label is based on the time the trajectory happened). Table 4 lists the results. By this comparison with SemanPredict, we can see that using semantic mining is inadequate for finding the relationship between the features we selected, and the accuracy of

Table 4. Comparison between different features.

Method	Features	Accuracy
SemanPredict	Semantic mining	53 % ~ 68 % (Affected by minimum support)
	Geographic mining	
Behavior semantic patterns mining of SemanPredict	Behavior semantic	63.1 %
	Transfer mode semantic	
	Behavior semantic	61.3 %
	Time semantic	
	Time semantic	40.1 %
	Transfer mode semantic	
	Behavior semantic	41.5 %
	Transfer mode semantic	
	Time semantic	

Table 5. Comparison between SemanPredict and TransemanPredict based on HMM.

Method	Accuracy
SemanPredict	53 % ~ 68 % (Affected by minimum support)
TransemanPredict based on HMM	68.3 %

SemanPredict methods is affected by minimum support. This is the reason why we use the HMM to try to obtain better results. Table 5 presents the results showing that HMM can be used according to our strategy with results better than those of SemanPredict.

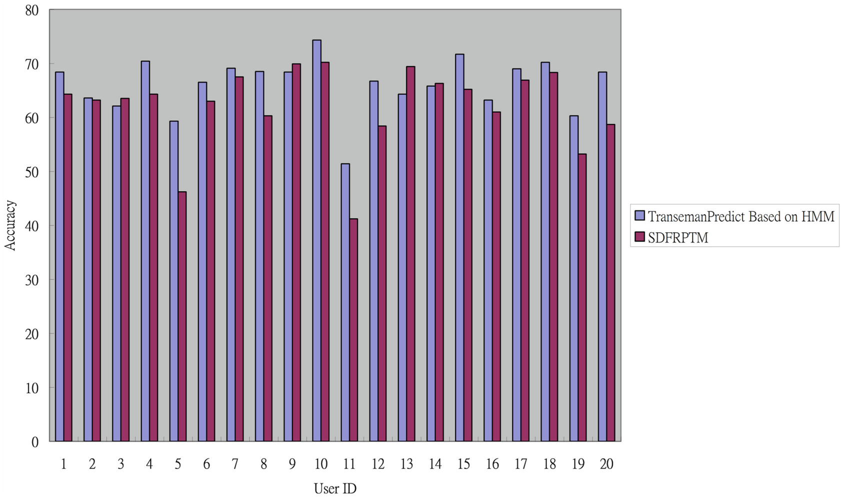


Fig. 6. Comparison between SDFRPTM and TransemanPredict based on HMM.

We conducted another comparative experiment for the prediction system comparing the accuracy between our prediction method and [25]. Figure 6 describes the results of the experiment in which we chose 20 users to demonstrate the performance of our prediction method. Our prediction method has a better accuracy than that of SDFRPTM for most of the mobile users.

4.4 Personal Semantic Label Generations

This section discusses the difference between using and not using the personal semantic labels. As can be seen in Table 6 describing the results, personal semantic label generation is proven capable of helping improve the precision of predicting the next location of the users.

Table 6. Comparison of the effects of using personal semantic labels.

	Accuracy	
	Google map semantic label	Google map semantic label + personal semantic label
TransemanPredict based on behavior semantic and Transfer situation semantic by semantic pattern mining	58.2 %	63.1 %
TransemanPredict based on HMM	67.5 %	68.3 %

5 Conclusion and Future Works

In this paper, we proposed an approach based on transportation mode and behavior semantic features to predict the next location of a user's movement. Our techniques predicted the next location in accordance with our common observation and experience in life, because humans often decide their destination by predictable desires and motivation. Through a serious of experiments, we demonstrated our location prediction strategy exhibits excellent performance.

Looking to the future, we plan to apply more features such as time, weather, traffic conditions and holidays etc. in order to increase the accuracy rate of our approach.

References

1. Microsoft's GEOLIFE project. http://reasearch.microsoft.com/en-us/projects/geolife/
2. Zheng, Y., Xie, X., Ma, W.Y.: GeoLife: a collaborative social networking service among user, location and trajectory. IEEE Data Eng. Bull. **33**(2), 32–39 (2010)
3. Nakahara, F., Murakami, T.: A destination prediction method based on behavioral pattern analysis extracting from nonperiodic position logs. In: ACM LBSN (2011)
4. Ying, J.J.C., Lee, W.C., Weng, T.C., Tseng, V.S.: Semantic trajectory mining for location prediction. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 34–43. ACM (2011)

5. Xie, R., Luo, J., Yue, Y., Li, Q., Zou, X.: Pattern mining, semantic label identification and movement prediction using mobile phone data. In: Zhou, S., Zhang, S., Karypis, G. (eds.) ADMA 2012. LNCS, vol. 7713, pp. 419–430. Springer, Heidelberg (2012)
6. Jeung, H., Shen, H.T., Zhou, X.: Mining trajectory patterns using hidden Markov models. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2007. LNCS, vol. 4654, pp. 470–480. Springer, Heidelberg (2007)
7. Mathew, W., Raposo, R., Martins, B.: Predicting future locations with hidden Markov models. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, pp. 911–918. ACM (2012)
8. Nguyen, N.T., Phung, D.Q., Venkatesh, S., Bui, H.: Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, CVPR 2005, vol. 2, pp. 955–960. IEEE (2005)
9. Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.Y.: Understanding mobility based on GPS data. In: Proceedings of the 10th International Conference on Ubiquitous Computing, pp. 312–321. ACM (2008)
10. Stenneth, L., Thompson, K., Stone, W., Alowibdi, J.: Automated transportation transfer detection using GPS enabled smartphones. In: 2012 15th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 802–807. IEEE (2012)
11. Trestian, I., Huguenin, K., Su, L., Kuzmanovic, A.: Transportation transfer detection using GPS enabled smartphone, In: International IEEE Conference on Intelligent Transportation Systems (2012)
12. Yan, Z., Chakraborty, D., Parent, C., Spaccapietra, S., Aberer, K.: SeMiTri: a framework for semantic annotation of heterogeneous trajectories. In: Proceedings of the 14th International Conference on Extending Database Technology, pp. 259–270. ACM (2011)
13. Yan, Z., Spaccapietra, S.: Towards semantic trajectory data analysis: a conceptual and computational approach. In: VLDB PhD Workshop (2009)
14. Lv, M., Chen, L., Chen, G.: Discovering personally semantic places from GPS trajectories. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 1552–1556. ACM (2012)
15. Chen, X., Pang, J., Xue, R.: Constructing and comparing user mobility profiles for location-based services. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, pp. 261–266. ACM (2013)
16. Monreale, A., Pinelli, F., Trasarti, R., Giannotti, F.: WhereNext: a location predictor on trajectory pattern mining. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 637–646. ACM (2009)
17. Ye, Y., Zheng, Y., Chen, Y., Feng, J., Xie, X.: Mining individual life pattern based on location history. In: Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09, pp. 1–10. IEEE (2009)
18. Blunsom, P.: Hidden Markov models. Lecture Notes, August 2004
19. Rabiner, L., Juang, B.: An introduction to hidden Markov models. IEEE ASSP Mag. **3**(1), 4–16 (1986)
20. Chen, Z., Sanjabi, B., Isa, D.: A location-based user movement prediction approach for Geolife project. Int. J. Comput. Eng. Res. **2**(7), 16–19 (2012)
21. Zheng, Y., Zhang, L., Ma, Z., Xie, X., Ma, W.Y.: Recommending friends and locations based on individual location history. ACM Trans. Web **5**(1), 5 (2011)
22. Google: Google Places API. http://developers.google.com/places/
23. Ying, J.J.C., Lu, E.H.C., Lee, W.C., Weng, T.C., Tseng, V.S.: Mining user similarity from semantic trajectories. In: Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, pp. 19–26. ACM (2010)

24. J48 Classifier. http://weka.sourceforge.net/doc/weak/classifiers/trees/J48.html
25. Chen, L., Lv, M., Chen, G.: A system for destination and future route prediction based on trajectory mining. Pervasive Mob. Comput. **6**(6), 657–676 (2010)
26. Lewis, D.D.: Naive (Bayes) at forty: the independence assumption in information retrieval. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 4–15. Springer, Heidelberg (1998)
27. Suykens, J.A., Vandewalle, J.: Least squares support vector machine classifiers. Neural Process. Lett. **9**(3), 293–300 (1999)

Query Prediction by Currently-Browsed Web Pages and Its Applications

Hui-Ju Hung[1(✉)] and Pu-Jen Cheng[2]

[1] Research Center for Information Technology Innovation & Institute
of Information Science, Academia Sinica, Taipei, Taiwan
hjhung@citi.sinica.edu.tw
[2] Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
pjcheng@csie.ntu.edu.tw

Abstract. This paper reveals the relation between a previously-browsed webpage and a query. First, we display that there are queries triggered by its previously-browsed webpage using real examples from the log. A query is triggered by a webpage means that the issued query is related to the webpage that the user had browsed before. Then an analysis is provided to show that almost 30 % of queries following a webpage are triggered. A predictor is proposed to detect the triggered queries. We also demonstrate that the predictor can be enhanced by giving previous queries as context. Finally, we show that the prediction can be applied on a query recommendation system to suggest queries for the currently-browsed web page.

Keywords: Trigger · Webpage · Prediction · Query recommendation · Context-aware

1 Introduction

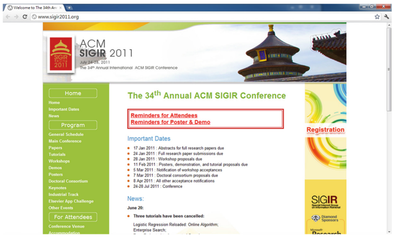
Nowadays, searching on the web becomes more and more challenging due to the unprecedented amount of data and small length of queries. Thus, understanding users' real intent below the query plays a key role in improving the quality of web search. Many commercial search engines provide query suggestion to predict users' intent based on previous issued queries. However, not only the previous query can be adopted as the materiel of query prediction, the previous viewed web page might also give users innovation to issue a related query. Few researches have paid attention to it.

A previous-viewed webpage may make a user to think of something, but the topic of what users might think of is very diverse. For example, suppose there is a user who has viewed the homepage of SIGIR'2011 (see Fig. 1) now [1]. After viewing the web page, the user might want to know more about the conference. Moreover, he or she might want to know more about the city where the conference was held, like the weather, the scenic spots, or the history of the city, etc. Another possibility for the user is seeking for the information of the building in the image. In addition, he or she might think of a SIGIR paper that he or she read before, or think of some other conferences like WWW or CIKM. There are many possibilities.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 181–193, 2014.
DOI: 10.1007/978-3-319-13186-3_17

However, not every notion leads to a query. We wonder whether there is any viewed web page that trigger user to issue a query. Unfortunately, only the original user who issued the query knows whether the query is triggered by the web page or not. Therefore we reasonably assume that a query is triggered if a user issues a related query after viewing a webpage. With this assumption, we observed that this phenomenon really exists in the log. For instance, a user viewed the Wikipedia page of Andy Carroll, an English footballer, and issued the query “Andy Carroll” right after viewing the page. Another example is that a user who viewed the Wikipedia page of American Craftsman, described as a style of domestic architectural design, and then issues the query “mayo furniture”, which is a furniture company in America.

Since the topics of notions from a webpage are diverse, the topics of triggered queries are diverse, too. Even an image or a short sentence might interest users to issue a corresponding query. Therefore, it is extremely hard to list out all the possible queries that will be triggered. In our work, we adopt a classifier trained by SVM to predict whether a specific web page will trigger a specific query. Then we show how the classifier could be applied to suggest queries when a user is browsing a web page.



The 34th Annual ACM SIGIR Conference

Reminders for Attendees
Reminders for Poster & Demo

Important Dates

- 17 Jan 2011 : Abstracts for full research papers due
- 24 Jan 2011 : Full research paper submissions due
- 28 Jan 2011 : Workshop proposals due
- 11 Feb 2011 : Posters, demonstration, and tutorial proposals due
- 5 Mar 2011 : Notification of workshop acceptances
- 7 Mar 2011 : Doctoral consortium proposals due
- 8 Apr 2011 : All other acceptance notifications
- 24-28 Jul 2011 : Conference

News:

June 20:

- Three tutorials have been cancelled:

Logistic Regression Reloaded: Online Algorithm;
Enterprise Search;

Fig. 1. Screenshot of the SIGIR 2011 web site (www.sigir2011.org)

2 Related Works

Many works have been done for understanding users’ intent below the query [2–4]. White et al. [2] paid attention to the website recommendation given the currently-browsed web page and five different types of contextual information—social, historic, task, collection and user interaction. More specifically, social combines the interests of other users that have browsed the currently-browsed webpage; historic is the interests of the current user; task contains the web pages which shares the same queries with currently-browsed webpage in search engines; collection contains the web pages which link to the currently-browsed webpage by hyperlinks; user interaction contains the current users’ recent interaction before browsing the currently-browsed webpage. Shen et al. [3] understood users’ intent from sequential search data using their model, Sparse Hidden-Dynamic Conditional Random Fields. Given a session of a server-side search log, which is composed of several search and click events, the model predicted which label the user’s intent in this session belongs to. In addition, eight different labels was defined and labeled in this work. Cheng et al. [4] displayed a prediction of real intent below users’ query. Then the prediction was used to rank queries following a given webpage for recommendation queries from the webpage. In addition, in their analysis, there are 23.8 % “browsing → search” patterns.

The contextual information has been widely used in information retrieval field. As mentioned, five kinds of contextual were adopted in [2]. Cao et al. [5] took the immediately preceding queries into account as context. There are two stages in the system. At the offline stage, queries in log were summarized into concepts and a concept sequence suffix tree was built by aggregating the sessions in log. At the online stage, the concept sequence suffix tree was adopted to suggest queries. He et al. [6] proposed a query recommendation method which predicts the next query by giving a sequence of preceding queries.

For recommendation, it is also possible way to retrieve important terms from the webpage. Therefore some woks [7–9] of summarization and key word identification are reviewed in the following. Sun et al. adopted both plain text and click through data to improve summarizing [7]. Grineva et al. [8] proposed a novel way to extract key terms by modeling a document (webpage) into a graph. In the graph, nodes represent terms, and edges represent the semantic relation between terms. Since the authors claimed that important terms tends to cluster with other terms densely. A graph community detection algorithm was adopted to partition the graph into sub-groups, and a criterion function is employed to select the group that contains key terms.

3 Problem and Methods

In this section, we describe each step in detail. Figure 2 shows the whole structure of our work. First, Trend Micro log is cleaned by removing those events with a URL directing users to a destination that is not a web page. Then we retrieve P → Q patterns, which P denote a webpage, and Q denote a query. Those P → Q patterns are labeled and cleaned by human judges to further remove illegal webpage and query and classify the patterns. Cleaned and labeled P → Q patterns are used to train the predictor with our features and get the result.

3.1 Problem Specification

Given a pair of a web page U and a query Q, representing that a user had viewed the web page and then issued the query subsequently, identify whether the query Q is triggered by the webpage U. It is a binary classification problem which classifies an $U \rightarrow Q$ pair into triggered type or non-triggered type (refer Table 2).

3.2 Data Set

The client-side log we used to retrieve users’ intent was crawled by Trend Micro. User actions observed at client side would be sent and recorded in the Trend Micro server. In this work, only those records of web browsing and searching histories were extracted. The log is recorded originally for security issues. That is, for each URL-viewing event, the URL would be sent to the server by anti-virus software to check

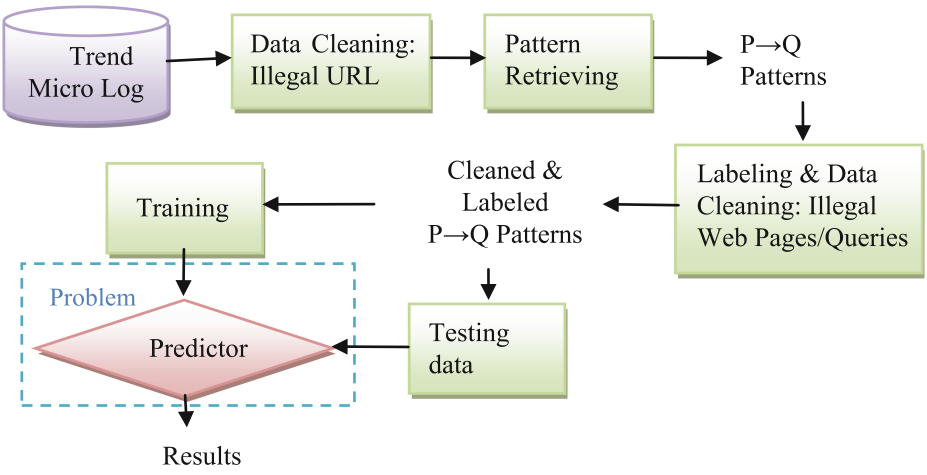

Trend Micro Log
Data Cleaning: Illegal URL
Pattern Retrieving
P→Q Patterns
Labeling & Data Cleaning: Illegal Web Pages/Queries
Cleaned & Labeled P→Q Patterns
Training
Problem
Predictor
Testing data
Results

Fig. 2. Workflow

whether the URL is safe or not. For a URL which has be checked and claimed to be safe in recent time, the anti-virus software may not send it to the server again. Thus the URL would be not recorded in the log.

However, The Trend Micro log recorded most actions in client computers which had installed the anti-virus software of Trend Micro. A one-hour log is collected from 0 A.M. to 1 AM., October 8, 2010. The languages used in this log are mainly English, Chinese, and Japanese.

Each entry in the log contains a unique user ID, timestamp, and the URL of a viewed web page. Those entries with a URL directing users to a Google search engine result page (SERP) are defined as query entries, denoted as Q, otherwise page-viewing entries, denoted as U. With the definition, a search section may look like

$$U \rightarrow U \rightarrow Q \rightarrow U \rightarrow Q \rightarrow Q \rightarrow U \rightarrow U\ldots$$

Note that a session ends if the user idles for more than thirty minutes. Moreover, we only collected those sessions with at least one search entry.

All sessions can be categorized into either query sessions or URL sessions. A Query session is a session that started with a query entry, and A URL session is a session starting with a page-viewing entry.

Table 1 shows the statistics of the one-hour log we use. In the table, the average number of terms in a query is 1.08298. Here the terms in a query are split by the original user using spaces or plus signs. No further word segmentation method is adopted.

As in Sect. 3.1, we focus on whether a query is triggered by its previously-viewed webpage. Thus, only URL sessions are adopted to train the predictor. The major reason is that URL sessions may capture users' behavior better when users are surfing on the Internet. Compared with URL sessions, the user behavior revealed in query sessions is more likely to attempt satisfying a pre-existing information need.

Table 1. Statistics of our dataset

Entity	Value
Number of sessions	1293
Number of query sessions	471
Number of URL sessions	822
Number of query events	6158
Number of URL events	37691
Number of unique queries	3365
Number of unique URLs	25043
Average number of query terms in a query	1.08298
Number of U → Q patterns in URL sessions	1907
Number of U → Q patterns in query sessions	353

3.3 Cleaning

Since the log recorded by Trend Micro contains almost every URLs that users accessed, there are many URLs that is not a real web page in the log, including document files (".doc", ".txt"), image files (".jpg", ".gif"), executable files, ad, frame, java script… etc. The major goal of cleaning is to remove URLs that are not linked to a web page.

There are two stages of cleaning in our work, cleaning illegal URLs and cleaning illegal web pages/queries. While in the stage of cleaning-illegal URLs, we removed the entries with URL that is impossible to be a web page. More precisely, we removed a URL that ended with a filename extension which is neither ".html" nor ".htm".

However, the rule is not effective enough to remove all entries in which the URL is not a webpage. Therefore, we adopted a further cleaning by human judge, called cleaning illegal web pages/queries. In this stage, we combine the labeling and advanced cleaning together. Firstly, a query and its previous web pages were retrieved until another query or the head of this session is reached. If there are more than five previous web pages, only the last five web pages are kept. After that, we got a pair composed of at most five URLs and a query. The judges were asked to select the latest valid URL. If there was no valid URL, the pair would be labeled as broken. In addition, a pair would be labeled as broken if its query contains nothing or can't be decoded into readable characters sequence.

3.4 Labeling

In this subsection we describe our labeling system. At the top, the ID and result of this pair is showed. Next, the five URLs, the query and the choice of relation are displayed. The judges are requested to select the latest valid URL, and then identify the relationship between the selected URL and the query. Each pair is classified by a judge. In this work, there are 5 judges aged between 22 and 26. Judges can easily view any content of a URL by simply clicking the link. In addition, since the query of a pair is retrieved from the URL of a Google search engine result page and there are not only

English queries in our dataset, some of queries are encoded by RFC 1738 [10], we also provide the query after decoding in the system to make judge easily.

For an unlabeled pair, both the selected URL and relationship is "have not been labeled". After being labeled, the selected URL must be one of URL1, URL2, URL3, URL4, or URL5. For a broken pair, the selected URL is labeled as one of URL1–URL5, and the relationship is still "have not been labeled". Unlabeled and broken pairs are excluded from the training and testing of our predictor. In our dataset, there are 1907 pairs from URL sessions. We labeled 1416 pairs from URL sessions, and 325 of them are broken.

Pairs that are neither unlabeled nor broken can be classified into three categories—triggered, non-triggered, and unknown. The labeling categories and instructions are showed in Table 2. Moreover, a triggered pair can be categorized into main topic or non-main topic; we will discuss the difference between them in Sect. 5.2.

Table 2. Labeling categories of a valid pair

Category	Instruction
Triggered	Q was issued because user had read U and felt interested in something in U
Non-triggered	Q was issued and U was read are independent events to each other
Unknown	This category include those pairs that cannot be classified into any of previous categories, usually because (1) Q cannot be understood by judges (2) the relation between P and Q is not clear

Most unknown pairs are labeled because the webpage that user had read require logging in to access more information. Another common reason is the query cannot be understood by judges. Some of queries contain just one letter of the English alphabet or a single number.

3.5 Basic Feature Engineering Method

After being labeled, a labeled pair contains a selected webpage U, a query Q, and the relation between them. In this section, 18 features are introduced and explained. In addition, since there are frequency features, we propose and compare three different ways to compute the "frequency" while there are multiple terms in a query in Sect. 0. Then, these features are employed to train and test by LIBSVM [11].

Features. First, the features used are listed in the following:

- F1: Rank (exist: 0–100, not exist:101)
- F2: Top 10? (yes: 1, no: 0)
- F3: Top 20? (yes: 1, no: 0)
- F4: Top 30? (yes: 1, no: 0)
- F5: Top 50? (yes: 1, no: 0)
- F6: Top 100? (yes: 1, no: 0)
- F7: Percentage of noun terms in query
- F8: Log of web page length in byte

- F9: Query frequency in the URL of web page
- F10: Query frequency in the webpage
- F11: Query frequency in the title of the webpage
- F12: Query frequency in the body the webpage
- F13: Query frequency in the first paragraph of the webpage
- F14: Query frequency in the last paragraph of the webpage
- F15: Query frequency in the first sentence of any paragraph of the webpage
- F16: Query frequency in the last sentence of any paragraph of the webpage
- F17: Percentage of the query terms that exists in the title of any Wikipedia articles
- F18: cosine similarity between the web page and the query

Among those features, F1 – F6 are ranking features, designed for capturing the similarity and relations strength between *U* and *Q*. When the user issues *Q*, the value of F1 is the rank of *U* if *U* is at the top 100 results, otherwise 101. F2 – F6 asked whether U is at the top 10/20/30/50/100 results when *Q* is issued. F7 reports the percentage of noun terms in *Q*. F8 is the logarithms of the length of *U* to base 2, which the length is measured by number of bytes.

F9 – F16 is frequency features. F9 reports the frequency of *Q* in the URL of *U*. F10 is the frequency of the *U*, including in the title. F11 is query frequency in title of webpage. F12 is the query frequency in the body of the webpage, and the HTML tag <body> stands as the borderline of HTML body part. F13 – F16 are the frequency of *Q* in first paragraph/last paragraph/first sentence in any paragraph/last sentence in any paragraph of *U*. These four features are proposed since descriptions that appear in above locations might be more important than in other locations. The html tag <p> is employed to separate different paragraphs, and the period (.) is adopted to separate sentences. F17 reports the percentages of terms in *Q* that exists in the title of any Wikipedia article. F17 is reported because the terms appeared in Wikipedia might be more difficult to understand or interesting, and make users tend to query it. F18 is the cosine similarity between term frequency (TF) 12 vectors of *Q* and *U*. Since most queries are short and sparse, the Google search result page of top 100 results is crawled for representing the query. Thus, F18 computes the cosine similarity between *U* and snippets of *Q*.

Handling Multiple Query Terms. Usually, a query contains several terms. It is hard to match the whole query in the document. Three different ways are presented to aim at this problem.

Consider a query Q = {t_1, t_2, t_3… t_n} and a document D,

- Maximum

$$\text{maximum} = \max_{t \in Q}(\text{frequency}_t(D))$$

- Average frequency

$$average\,frequery = \frac{\sum_{t \in Q} \text{frequency}_t(D)}{n}$$

- Average appearing possibility

$$average\,apperaing\,possibility = \frac{\sum_{t \in Q} \mathrm{I}_t(\mathrm{D})}{n}$$

Where

$$\mathbf{I}_t(\mathbf{D}) = \begin{cases} \mathbf{1},\ \boldsymbol{frequency_q}(\boldsymbol{D}) > 0 \\ \mathbf{0},\ \boldsymbol{frequency_q}(\boldsymbol{D}) = \mathbf{0} \end{cases}$$

These three methods for computing the frequency have been evaluated on the dataset. However, the accuracies of them are almost the same. Among them, the maximum method performs slightly better than others. Therefore, the maximum method is adopted in the following experiment.

3.6 Context-Aware Method

In this method, previous queries are considered. For the query of each pair, the last previous query within the same session is detected, called *Q*'. Pairs without *Q*' (i.e. the query is first query entry in the session) are discarded. Subsequently, one new feature is added.

- F19: cosine similarity (*Q*, *Q*')

Aiming at the sparseness of queries, similarly as mentioned in Sect. 0, the Google search result page of top 100 results are crawled. The term frequency vector of the page is adopted as the vector of the query.

3.7 Evaluation

To evaluate the performance of the predictor, the accuracy measurement is employed. Moreover, 5-fold cross validation is performed to validate the accuracy.

The accuracy is defined in the following:

$$\text{accuracy} = \frac{\#tp + \#tn}{\#tp + \#fp + \#tn + \#fn}$$

Where

		Gold standard (Labeling result)	
		Triggered	Non-triggered
Predicting result	Triggered	tp	fp
	Non-triggered	fn	tn

4 Experimental Results

4.1 Experiment Setting

The statistic of labeled results is shown in Table 3. Labeled result distribution

Table 3. Labeled result distribution

Category	# of patterns	Percentage	Percentage(exclude “Broken”)
Trigger	296	20.90	27.13
Non-Trigger	694	49.01	63.61
Unknown	101	7.13	9.26
Broken	325	22.96	
All	1416		

For balancing the number of pairs from triggered and non-triggered, we sampled 100 triggered pairs and 100 non-triggered pairs as our data set.

4.2 Results of Basic Feature Engineering Method

Baseline. The baseline adopted is described as follows: If the percentage of appearing query terms is larger than a threshold *T*, the pair belongs to triggered type, otherwise it belongs to non-triggered type. Table 4 provides the baseline accuracy in different threshold.

Table 4. Baseline result of selected data set at different threshold

T	0	0.1	0.2	0.3	0.4	0.5
Accuracy	67.5	67.5	67.5	67	66.5	67
T	0.6	0.7	0.8	0.9	1	(%)
Accuracy	66.5	66.5	63	62.5	0.5	

With T is equal of 20 %, the accuracy has a peak at 67.5 %. Thus T is set to 20 % and the accuracy of the baseline is 67.5 %.

Accuracy. Table 5 displays the accuracy of the basic feature engineering method compared with the baseline.

Therefore, we get 17.78 % improvement from baseline using following formula.

Table 5. Accuracy of basic feature engineering method

Method	Basic feature engineering method	Baseline
Accuracy (%)	79.5	67.5

$$improvement = \frac{method\ accuracy - baseline\ accuracy}{baseline\ accuracy}$$

Feature Effectiveness. For testing the effectiveness of each feature, we remove every feature one by one, and use the accuracy change (AC) to measure the effectiveness.

$$Accuracy\ Change\ (F) = \frac{feature\ accuracy - overall\ accuracy}{overall\ accuracy}$$

Feature accuracy change denotes the new accuracy after removing the exactly one feature F that we want to test. Overall accuracy denotes the accuracy with all features (Table 6).

Table 6. Feature effectiveness

Feature	AC	Feature	AC
cosine similarity(U,Q)	−6.92 %	Top 20	−0.63 %
Q freq. in URL of U	−6.29 %	Top 100	−0.63 %
log2 (web page length)	−3.77 %	Rank	0.00 %
Q freq. in U	−3.77 %	Top 10	0.00 %
Q freq. in last sentence of U	−3.14 %	Top 30	0.00 %
Q freq. in first paragraph of U	−2.52 %	Top 50	0.00 %
Q freq. in <body> of U	−1.26 %	Q freq. in <title> of U	0.00 %
Q freq. in last paragraph of U	−1.26 %	% of noun terms	0.63 %

4.3 The Context-Aware Method

Baseline. The rules of baseline are described here. If the percentage of appearing query terms is larger than the threshold T, the pair belongs to triggered type, otherwise it belongs to non-triggered type. With T is equal of 20 %, the best baseline accuracy is reached at 62.86 %. Thus T = 20 % is selected (Table 7).

Table 7. Baseline result of context-aware method at different threshold

T	0	0.1	0.2	0.3	0.4	0.5
Accuracy	62.86	62.86	62.86	61.90	59.05	58.10
T	0.6	0.7	0.8	0.9	1	(%)
Accuracy	58.10	59.05	55.24	54.29	54.29	

Table 8. Accuracy of context-aware method

Method	Context-aware method	Baseline
Accuracy (%)	83.81	62.86

Accuracy. A 33.33 % improvement from baseline is reached, and a 5.42 % improvement from basic feature engineering method is reached (Table 8).

5 Discussions

5.1 Feature Effectiveness in Basic Feature Engineering Method

The aim of this research is to study the relation between a query and its previous webpage, and to predict the relation between a query and a webpage in an unknown pair. Firstly, the statistics of labeling result indicate that almost 30 % of $U \rightarrow Q$ pairs are triggered, which is large enough to be focused on. Secondly, with the 18 features in basic feature engineering method, there is a significant 17.78 % improvement in accuracy from 67.5 % to 79.5 % comparing to the baseline.

Most frequency features contribute to the predictor except F11. Recalled the assumption of triggered $U \rightarrow Q$ in this work is that a user viewed U and issued a related Q. Therefore if Q appeared in U, the judges can easily find the relationships in most cases. Thus F8, the query frequency in the webpage, performs well in the predictor. However, among these frequency features, F9, the query frequency in URL of the web page, performs the best. The reason might be that there are some pairs in which the user issued the domain name as the query when browsing a web page. One possible explanation is the user would like to access the homepage but the hyperlink is difficult to find. In addition, there are some location features (F9, F11, and F13–F16) that focus on where the query term appears in the content of the webpage. F13–F16, designed to capture the frequency in specific location, performs worse than matching the whole webpage (F10), but F13 and F16 performs better than matching the content of the webpage (F12). Unexpectedly, F11, the query frequency in the title of the webpage, does not work well. Because the title of a webpage may be usually short, it is hard to match terms in the title.

Ranking features, originally designed for capturing the strength of relationship between the query and the webpage, are too sparse to be adopted since only top 100 were checked. Other web pages are treated as non-relevant web pages even if the rank is very close to 100.

Percentage of noun terms in the query is not effective since most of query terms are noun no matter whether the query is triggered or not. Moreover, whether a query is triggered is decided by not only the query, but also the webpage. Two pairs with the same query may also be in different types.

Percentage of wiki terms in the query is unfortunately not effective, too. Since no advanced word segmentation were applied, it is possible that a query term is not in title of any Wikipedia articles but its substring is, like “NTUCSIE”. (That is also the reason that whether the whole query is the title of any Wikipedia articles is not adopted as the feature.) Even worse, it is also possible that every term appeared in some Wikipedia articles but the appearing is not related to the topic of the query. For example, there is a real query "Dr. Randolph Capri" in the log, who is a doctor working in California. In fact, there is no Wikipedia article written about him. However, "Randolph" and "Capri" both appear in some titles of Wikipedia articles; therefore the percentage is 66.67 %. This feature introduced too many noises.

5.2 Main Topic and Non-main Topic

The triggered category can be divided into two sub-categories – main topic and non-main topic. If the triggered query is the main topic of the web page, this pair belongs to main topic; otherwise it belongs to non-main topic.

Non-main topics are harder to detect since the frequency may be very low, and the fields of triggered queries may be diverse. Sometimes the query does not appear in the webpage but the query is stilled triggered.

In our dataset, the 100 triggered patterns are composed of 90 main topic patterns and 10 non-main topic patterns.

5.3 Context-Aware Method

Given the last previous query *Q*' information, we assume the webpage *U* is the search result of *Q*'. The order that user viewed is $Q' \rightarrow U \rightarrow Q$. With *Q*', we could predict those patterns whose similarity between *U* and *Q* are small but they are possibly triggered. Therefore, the accuracy is improved by giving the information of previous query.

6 Conclusions

In this paper, we proposed a prediction of triggered queries based on a real client side log. The predictor combines the frequency, the similarity, the webpage length and the rank as features. Among the above features, the frequency and similarity features outperform from others. The locations features also contribute to the accuracy of prediction. Then, the prediction is enhanced by giving the previous queries as context. Moreover, many application may be done by our prediction, such like query suggestions on mobile devices, AD generation for blogs, and implicit hyperlink construction for web structure.

References

1. http://www.sigir2011.org/
2. White, R. W., Bailey, P., Chen, L.: Predicting user interests from contextual information. In: SIGIR (2009)
3. Shen, Y., Yan, J., Yan, S., Ji, L., Liu, N., Chen, Z.: Sparse hidden-dynamics conditional random fields for user intent understanding. In: WWW (2011)

4. Cheng, Z., Gao, B., Liu, T.-Y.: Actively predicting diverse search intent from user browsing behaviors. In: WWW (2010)
5. Cao, H., Jiang, D., Pei, J., He, Q., Liao, Z., Chen, E., Li, H.: Context-aware query suggestion by mining click-through and session data. In: KDD (2008)
6. He, Q., Jiang, D., Liao, Z., Hoi, S. C. H., Chang, K., Lim, E.-P., Li, H.: Web query recommendation via sequential query prediction. In: ICDE (2009)
7. Sun, J.-T., Shen, D., Zeng, H.-J., Yang, Q., Lu, Y., Chen, Z.: Web-page summarization using clickthrough data. In: SIGIR (2005)
8. Grineva, M., Grinev, M., Lizorkin, D.: Extracting key terms from noisy and multi-theme documents. In: WWW (2009)
9. Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., Nevill-Manning, C. G.: KEA: practical automatic keyphrase extraction. In: DL (1999)
10. http://datatracker.ietf.org/doc/rfc1738
11. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 1–27 (2011). Article 27
12. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)

An Emergency System in Mobile Devices

Chung-Hua Chu(✉) and Wen-Hsiang Chang

Department of Multimedia Design,
National Taichung University of Science and Technology, Taichung, Taiwan
chchu777@gmail.com

Abstract. Mobile value-added services for smart phones enable users to enrich their daily lives. Recently, emergency rescue systems have been becoming imperative. However, the user interface of the traditional emergency rescue system is too, and many essential functions are not designed. This paper proposes an emergency rescue system on mobile devices. Mobile users can be fast rescued by sending SOS messages in SMS (Short Message Service) to their friends or family. Additionally, the SOS messages can also be posted on Social network such as Facebook. Specifically, the users can shake their mobile devices to send the SOS messages. Moreover, GPS information is contained in the SOS message such that the rescuers can fast and precisely find the location of the users. Therefore, the proposed framework is practical and necessary for the users to confront the future disasters.

Keywords: Emergency rescue system · Mobile device

1 Introduction

In recent years, smart phones have become popular, due to the advantages of user-friendly interface and diversified functions. The user population spans across all ages, even the elderly who are normally unfamiliar with technological products are willing to use smart phones. The market share of smart phones has increased continuously due to diversified and convenient value-added functions. The network platforms of mobile service providers offer thousands of apps for users to download, making smart phones to be more than just a tool for phone calls.

In recent years, natural disasters have occurred frequently around the world; hence, timely rescue has become important. If the locations of the disaster and persons in distress can be known instantly when a disaster occurs, the rescue search for victims can be accelerated. A device that that can send SOS signal instantly in accidents could save numerous lives. This study intends to utilize the acceleration sensor and satellite positioning of smart phone to develop such a real-time emergency system, in order to enhance the safety of people during disasters.

2 Related Works

The application of acceleration sensor creates an interactive mode for users and electronic products. Many interactions can be implemented by hand gestures. However,

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 194–198, 2014.
DOI: 10.1007/978-3-319-13186-3_18

it is a technical challenge for electronic products and mobile devices to recognize all hand gestures [3]. As the mobile devices are portable, each movement that the user makes generates a feedback of acceleration sensing, such as each action of walk and run [1]. However, the developers must design the conventional gestures of users in daily life correctly when designing the gestures related to mobile devices [2]. Ruiz proposed 14 habitual gestures of users in life.

3 Proposed Methods

3.1 System Architecture

The emergency system architecture of this study is shown in Fig. 1. The proposed system uses the acceleration sensor of smart phones to detect the acceleration, and the GPS positioning is detected when an acceleration threshold is reached. The latitude and longitude are converted into address message via network, and the address is sent out via Facebook and short message.

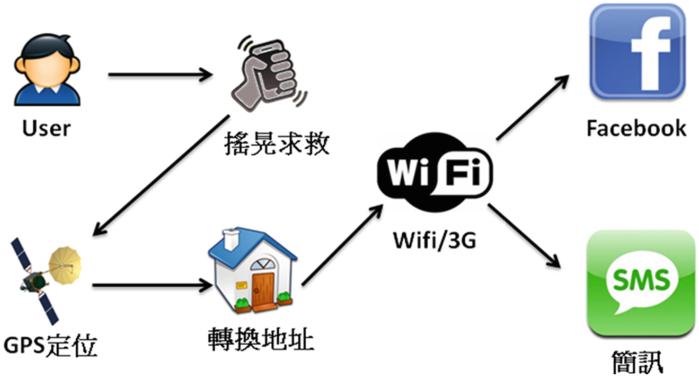


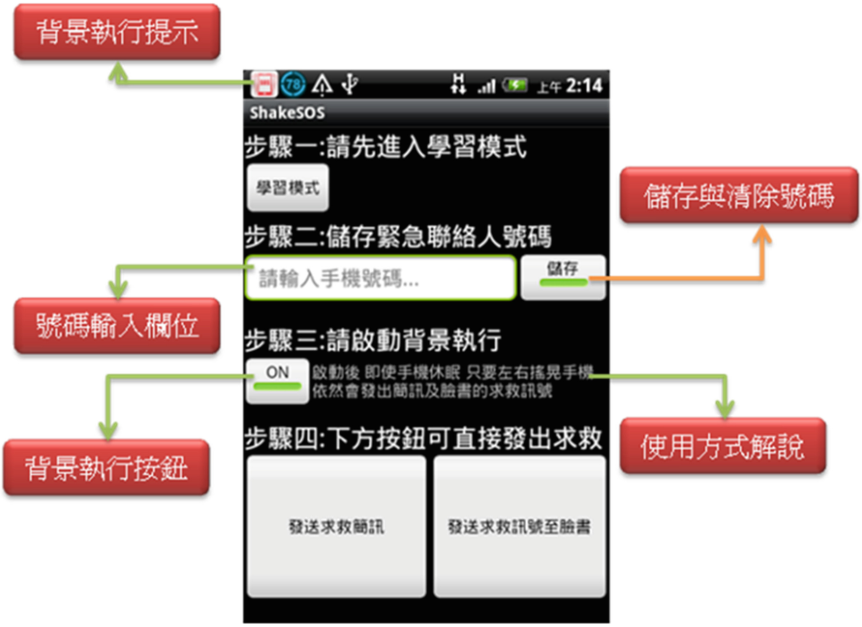
Fig. 1. System architecture

3.2 System Flow

Before use, the user is required to set related data, such as Facebook login and emergency contact numbers, and decide whether to start up background execution. When the system is turned on, it begins to detect acceleration; as long as the user does not stop background execution, the system keeps detecting acceleration; even the phone enters the sleep mode. If the acceleration exceeds the threshold, the system determines that the user has shaken the smart phone to ask for help, and then the GPS positioning is detected. When the latitude and longitude are obtained, the information is converted via network into a location address, which is posted on the user’s Facebook, and the short message is sent to the emergency number.

3.3 User Interface Design

Figure 2 shows the system operation icons. The clearly explained steps allow the users to learn the operation quickly. Step 1 is the learning mode, where the users learn to shake the smart phone; Step 2 asks the users to enter emergency contact number, while the same button is provided for Save or Clear, so that the users are aware of whether the contact number is saved successfully, and know how to clear the contact number; Step 3 shows the on-off button, and explains the purpose of background execution and operation, so that users do not have to learn the terminology; Step 4 provides the second SOS mode, besides the shaking SOS, the user can touch the button to send SOS signal if possible. This system provides Facebook and short message buttons, so that the users can select the sending mode as required.



ShakeSOS

步驟一:請先進入學習模式

步驟二:儲存緊急聯絡人號碼

請輸入手機號碼...

步驟三:請啟動背景執行

啟動後 即使手機休眠 只要左右搖晃手機
依然會發出簡訊及臉書的求救訊號

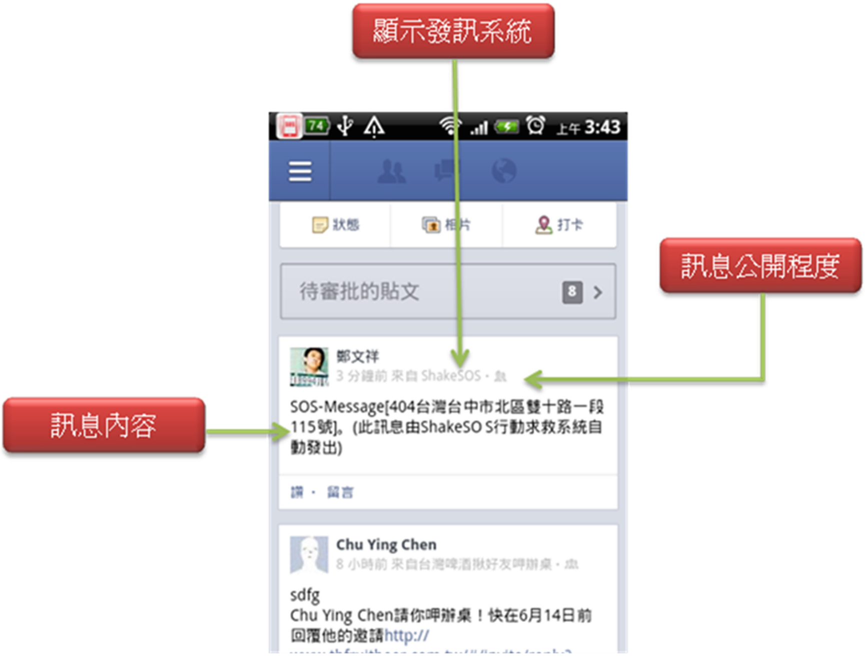
步驟四:下方按鈕可直接發出求救

Fig. 2. System user interface

Figure 3 shows the SOS message content. When a user sends out the SOS message to Facebook, the system releases the SOS message containing the address of the user to the Facebook friends in the extent of disclosure set by the user.

4 Experiments

In order to avoid the false triggering of SOS in normal actions, this study conducted tests using common actions in daily life, which were divided into three types based on the terrains: flat ground, stepping upstairs and downstairs. There are two behavior



鄭文祥
3 分鐘前 來自 ShakeSOS ·

SOS-Message[404台灣台中市北區雙十路一段115號]。(此訊息由ShakeSO S行動求救系統自動發出)

Chu Ying Chen
8 小時前 來自台灣啤酒揪好友呷辦桌 ·

sdfg
Chu Ying Chen請你呷辦桌！快在6月14日前回覆他的邀請http://

Fig. 3. SOS message content

actions, walk and run. According to pretest questionnaire result, most users place their smart phones in the pocket or bag, so the smart phone was also placed in those positions during test. Based on the principle of acceleration sensing, the placement of smart phone was divided into on upright, on lateral side, and lay flat (as shown in Fig. 4). The peak accelerations in X, Y and Z axes were obtained to simulate the phone placements in daily life. However, as the phone cannot be laid flat in the pocket, it is held in hand to simulate the situation that the user uses the phone when walking. Moreover, there is certain degree of vibration when the user is riding a motorcycle or when the phone is dropped, those two situations were also included in the test. The test results are shown as follows.

The stepping upstairs test was conducted at Changming Building, Taichung University of Science and Technology. The user walked four stories, and ran four stories. The maximum acceleration occurred in the running state during the test, and the smart phone was laid flat in the bag. The test results are shown in Table 1.

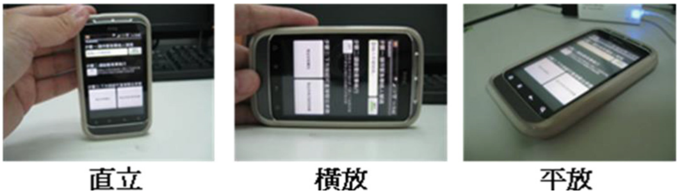

直立 橫放 平放

Fig. 4. Schematic diagram of placement of the smart phone

Table 1. Stepping upstairs acceleration test

State	Position	Placement mode	X maximum value	Y maximum value	Z maximum value
Walking	Pocket	Upright	11.372	18.836	12.707
		On lateral side	13.552	7.382	13.211
	Hand	Lay flat	4.167	3.282	17.311
	Bag	Upright	4.508	16.998	12.748
		On lateral side	12.669	7.504	11.863
		Lay flat	12.285	5.516	16.698
Running	Pocket	Upright	7.886	15.050	14.287
		On lateral side	12.871	10.800	14.818
	Hand	Lay flat	5.162	5.734	19.613
	Bag	Upright	5.625	11.481	10.950
		On lateral side	13.742	6.088	15.009
		Lay flat	20.144	19.449	19.613

5 Conclusion

The user demand for mobile emergency system and the use of smart phone shaking function were explored in the pretest. The acceleration was tested and the system was designed according to the preset result. The GPS latitude and longitude were converted into addresses as the SOS content by of the smart mobile phone background execution, and short messages were sent to the emergency contact number and the sender's Facebook. The post-test on user satisfaction found that the averages mean of questionnaire items is higher than 3, proving that the system is accepted by most users.

Acknowledgment. The work was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC102-2221-E-025-005.

References

1. Jedrzejczyk, L., Price, B.A., Bandara, A., Nuseibeh, B.: "Privacy-shake": a haptic interface for managing privacy settings in mobile location sharing applications. Paper presented at the proceedings of the 12th international conference on human computer interaction with mobile devices and services, Lisbon, Portugal (2010)
2. Ruiz, J., Li, Y., Lank, E.: User-defined motion gestures for mobile interaction. Paper presented at the proceedings of the 2011 annual conference on human factors in computing systems, Vancouver, BC, Canada (2011)
3. Liu, J., Wang, Z., Zhong, L., et al.: uWave: accelerometer-based personalized gesture recognition and its applications. Paper presented at the 2009 IEEE international conference on pervasive computing and communications (2009)

Study of Mobile Value-Added Services with Augmented Reality

Chung-Hua Chu(✉), Shu-Lin Wang, and Bi-Chi Tseng

Department of Multimedia Design,
National Taichung University of Science and Technology, Taichung, Taiwan
chchu777@gmail.com

Abstract. The research which combined Augmented Reality with technique of Assisted Global Positioning System (AGPS) constructed a guiding system of Augmented Reality and designed guiding graphs metaphorically; thus, the system interface operation was used more intuitively. The research further investigated the availability of the system, and an empirical study statistically showed that a guiding system of Augmented Reality significantly outperformed that of plane map in terms of finishing time of mission and correctness. Finally, according to the results of questionnaire of the system availability, the study inducted six essential factors influencing the guiding system availability of augmented reality, including guiding service usability factor, user aesthetics of design factor, guiding service technique factor, guiding service creativity factor, guiding service entertainment factor, and guiding service practicality factor. The results of the study could be referred to other related studies.

Keywords: Augmented reality · Mobile device

1 Introduction

Based on the above discussion, this study attempts to use a human-machine interface design to overcome the abovementioned defects in traditional navigation application. AR is used to provide an interactive interface for users, and improve the effect of urban navigation or other navigation services. Metaphorical elements are added by icon design, so as to make the interface operation more intuitive. In addition, this study adopts assisted global positioning system (AGPS) to enhance the timely effectiveness of positioning. The difference between AGPS and GPS is that the AGPS can use both the signals of mobile phone bases and GPS satellite signals to accelerate the positioning process. These modes enable the system to guide the users to the destinations accurately.

This study uses an iPhone for experiment. Based on the built-in digital compass and AGPS, this study adds the AR navigation system, which allows users to find out the important landmarks and tourist attractions in the cities, and displays relevant information about the destination. The navigation icons guide the users to reach the destination correctly. The metaphorical icons change in sizes and angles to allow the users to know the distance and direction to the destination.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 199–203, 2014.
DOI: 10.1007/978-3-319-13186-3_19

2 Related Works

AR refers to “the user wears a transparent display equipment on head which can fuse the scene of real world with the computer generated image directly; it is still a special VR in essence” [2]. However, VR differs from AR according to the degree of fusion. AR provides a composite landscape; the scene seen by the users contains both reality and virtuality. VR is a totally immersive environment, where the users’ vision, hearing and perception must be completely under the control of VR system. On the other hand, AR enables users to see the virtual objects overlapped in real environment, and enhances the reality instead of replacing the reality [1]. Azuma et al. proposed three necessary attributes of AR.

3 Proposed Methods

3.1 Mobile Navigation Type – AR

This mobile navigation system provides the navigation system suitable for mobile devices, and it integrates AR, metaphorically designed icons, and APGS, allowing the users to reach the destination easily by using the system. The proposed mobile navigation system is built by Objective-C, and is operated on iPhone. The users need to connect wireless network or 3G network to operate the system. The flow chart of AR navigation is shown in Fig. 1.

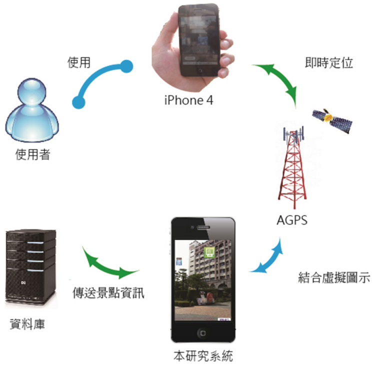


Fig. 1. Flow chart of AR navigation

3.2 Description of Interface Planning for AR Navigation

In order to ensure user-friendliness, the design of the system interface considers the following features: designed for mobile users, consistency, providing feedback, using metaphor, using icons to clarify concepts, proximity, similarity and providing appropriate text.

3.3 Site Information Display Interface Planning

The characteristics and implementation mode of interface for geo information of this system are observed. The site information is displayed in the content area, and the lower buttons are pressed by touching.

3.4 Context Display of AR Navigation

Figure 2 shows the context of AR navigation for NTCUST. A user is heading for the destination, guided by iPhone. First, this system is started, the screen displays camera mode, and the user uses the phone to observe the environment to look for the direction and location of destination. The picture combines real scene with virtual icons, and the virtual icons guide the user to the destination. The user presses the department information button to view the target information. The picture is switched to the department information page, and the information of the site is displayed. The user can select the information page of nearby sites.

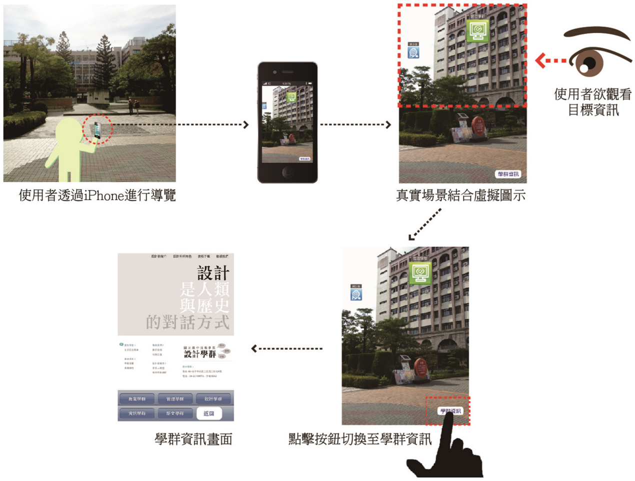


Fig. 2. Context of AR navigation

4 Experimental Design

4.1 Statistical Analysis

This study randomly invited 50 subjects to participate in the experiment, and 45 valid samples were collected. Most of the participants were not students of NTCUST, and were unfamiliar with the campus environment. In the experimental process, the participants were required to complete the navigation tasks of this experiment by using the navigation service function of the proposed system. The participants were required to fill out an evaluation questionnaire after the experiment.

4.2 System Performance Analysis

Reliability Analysis. In order to confirm the stability and effectiveness of the questionnaire, all the items and results were collected for reliability analysis. As shown in Table 1, the reliability coefficient is 0.944, indicating a high consistency as it is very close to 1. The Cronbach α value of the overall questionnaire items and dimensions are 0.7 to 1, indicating a high reliability.

Table 1. Reliability statistics of questionnaire items

Dimension	Cronbach's Alpha value	Number of items
Technicality	.798	4
Usability	.858	4
Innovation	.802	3
Applicability	.861	4
Design aesthetics	.821	3
Entertainment	.841	3
Intention to use	.849	2
Total	.946	23

Validity Analysis. As the research variables in this study were based on previous literature, the content was reviewed by ANOVA, and the dimensions were extracted by factor analysis. The results confirmed good content validity.

5 Conclusions

This study metaphorically designed an AR navigation system, and tested whether the navigation mode has a positive effect on the users. The experimental groups included the AR navigation group, the plane map (plain text) navigation group, and the plan map (text and icon) group. The results indicated in the plane map navigation mode, the participants needed to compare the surrounding with the images or text, so as to identify the location and direction. This process is time-consuming and has a high error rate. On the contrary, in the AR navigation mode, the users can know the correct direction and location immediately under the guide of virtual icons without the need to identify the surrounding. The process is efficient and the error rate is low. Therefore, the AR combined with reality and virtuality can accelerate the users' identification, and the real-time interaction enables the users to identify the current location immediately.

Acknowledgment. The work was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC102-2221-E-025-005.

References

1. Azuma, R.T.: A survey of augmented reality. Presence-Teleoper. Virtual Environ. **6**, 355–385 (1997)
2. Milgram, P., Takemura, H., Utsumi, A., Kishino, F.: Effects of fuzziness in perception of stereoscopically presented virtual object locations. In: Proceedings of SPIE Telemanipulator and Telepresence Technologies, vol. 2351-39 (1994)

Mining Uncertain Sequence Data on Hadoop Platform

Zi-Yun Sun, Ming-Che Tsai, and Hsiao-Ping Tsai(✉)

Department of Electrical Engineering, National Chung Hsing University,
Taichung, Taiwan, Republic of China
hptsai@nchu.edu.tw

Abstract. Sequence pattern mining is the mining of special and representative features hidden in sequence data. Recently, it has been attracting a lot of attention, especially in the fields of bioinformatics and spatio-temporal trajectory mining. Observing that many sequence data are born with uncertainties and huge sequence data are increasingly generated and accumulated, this paper aims to discover the hidden features from a large amount of uncertain sequence data. Specifically, Probabilistic Suffix Tree (PST) is an implementation of Variable-length Markov Chain (VMM) that has been widely applied in sequence data mining. However, the conventional PST construction algorithm is not for the mining of uncertain data and cannot bear the computing of huge data. Thus, to mine a large amount of sequence data with uncertainties, this paper proposes the uPST$^{+}_{MR}$ algorithm on the Hadoop platform to fully utilize the computing power and storage capacity of cloud computing. The proposed uPST$^{+}_{MR}$ algorithm constructs a PST in a progressive, multi-layered, and iterative manner so as to avoid excessive learning patterns and balance the overhead of distributed computing. In addition, to prevent the drag on overall performance owing to multiple scanning of the entire sequence data, we trade space for time by using a NodeArray data structure to store the intermediate statistical results to reduce disk I/O. To verify the performance of uPST$^{+}_{MR}$, we conduct several experiments. The experimental results show that uPST$^{+}_{MR}$ outperforms the naive approach significantly and show good scalability and stability. Also, although using NodeArray costs a little extra memory, the execution time is significantly lowered.

1 Introduction

Recent advances in wireless network, positioning technologies and mobile devices have driven the development of many mobile applications, e.g., wild animal and vehicle tracking, military surveillance, secure care of the elderly and children, mobile social network, etc. These applications quickly produce large amounts of data with the location coordinates and time field, which are thus called spatio-temporal data. These spatio-temporal data contain a lot of important knowledge and are very worthy of further research and exploration. The analysis of such

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 204–215, 2014.
DOI: 10.1007/978-3-319-13186-3_20

spatio-temporal data is also one of the most popular research directions in the field of data mining [3,4].

Many phenomena in nature suggest that the biological behaviour often has a certain degree of regularity, e.g., the route from home to office is of high regularity for many workers. Also, animals like elephants and whales have seasonal migratory behaviors along a particular trajectory, moving in a fairly regular type [1,2]. To analyse the movement behaviors, the first step is to obtain the location sequences of moving objects, which can done by tracking and continuously recording the position of moving objects over time. Observing that the sequence data inherently have some uncertainties, which may be caused by limitations of measurement techniques, sampling error, and privacy preserving [5–9]. Without considering the uncertainties and applying existing mining approaches on the uncertain data, the mining results may lose many details and features. Therefore, it is urgent to define new movement patterns according to the characteristics of uncertain sequence data to enhance the accuracy and precision of the discovered knowledge. Regarding uncertain trajectory sequence data, some researches have been recently published, focusing on uncertain trajectory data modelling and uncertain data management, or the exploration of popular trajectories of uncertainty [10,11]. However, studies on the mining of movement patterns with the consideration of the location uncertainty are rare. Only Leung et al. concern the uncertainty in mining sequential patterns [12]. However, due to the lack of continuity limitation of sequential patterns, the mining results cannot be used in predicting the location of near future directly.

Probabilistic Suffix Trees (PST) [13] is an implementation of VMM that has powerful capability of automatically capturing the data structural relationships. It is wildly used and has been proven to be useful in the patterns mining with high efficiency and good predictive ability. However, the conventional PST construction algorithm does not apply to the uncertain data. Moreover, since it is a centralized algorithm, it cannot bear the computing of huge data. Thus, to handle massive data computation and storage, a distributed approach is demanded. Recently, cloud computing has emerged to bring about new opportunities for information storage, processing and mining issues. It is a distributed computing technology and can be regarded as a realization of distributed computing. With a high degree of scalability and fault tolerance, it provides great computing power and huge storage capacity. The open source Hadoop by Apache is currently the most widely adopted cloud computing platform. Hadoop adopts MapReduce distributed programming framework, making the conventional algorithms no longer applicable.

For efficient mining of a large amount of sequence data with uncertainties, in this paper, we redefine the movement patterns and propose the uPST$_{MR}$ and uPST$^{+}_{MR}$ algorithms in line with MapReduce programming model to utilize the massive computing resources of the cloud computing. The uPST$_{MR}$ and uPST$^{+}_{MR}$ algorithms construct PST in a progressive, multi-layered, and iterative way. Thus, it can avoid excessive mining of patterns while balancing the distributed computing overhead. Moreover, to avoid the drag on the overall performance by multiple scanning of the entire sequence data, the uPST$^{+}_{MR}$ algorithm further uses a newly designed data structure for the temporary storage of intermediate statistical results. Therefore, the uPST$^{+}_{MR}$ algorithm only needs to scan the entire sequence data once in each MapReduce iteration to construct the specified PST layers. In order to verify the performance, this study carried out a number of experiments. The test results show uPST$^{+}_{MR}$ is better than uPST$_{MR}$ in all aspects. uPST$_{MR}$ can considerably reduce time consumption by the progressive, multi-layered and iterative way, and thus reducing the overall computing time substantially. uPST$^{+}_{MR}$ adopts the strategy of trading appropriate space for time. Therefore, when there are a lot of patterns in the data, the overall computing time will slowly increase, presenting a good scalability and stability.

The paper is organized as follows: Sect. 2 gives the preliminary about PST/VMM and definitions. Section 3 elaborates on the proposed algorithms. The experimental results are illustrated in the Sect. 4. Finally, Sect. 5 concludes the paper.

2 Preliminary

2.1 VMM/PST

VMM is a variation of Markov Model. Under the assumption of Variable Length Markov Model (VMM), the features such as the movement regularity in trajectory sequences are expressed by the conditional probability distribution. Specifically, for a given sequence dataset over an alphabet Σ, a subsequence s over Σ, and a symbol $\sigma \in \Sigma$, $p(\sigma|s)$ is the conditional probability that σ will follow s. The conditional probability distribution of s is expressed by $p(\sigma|s), \forall\sigma \in \Sigma$. The length of s is floating that provides flexibility to adapt to the variable length of patterns.

Let $S[i : j]$ denote the substring of S starting from i and ending at j. An occurrence of s at i of S is denoted by $I(S_i, s) = 1$; otherwise $I(S_i, s) = 0$. The number of times a subsequence s is observed in the sequence dataset S is $\sum_{i=0}^{|S|-|s|} I(S_i, s)$; the total number of overlapping sub-sequences of length $|s|$ in the dataset is $|S| - |s| + 1$; then, the empirical probability of a subsequence s over the given dataset S is $p(s) = \frac{\sum_{i=0}^{|S|-|s|} I(S_i,s)}{|S|-|s|+1}$. Moreover, the conditional empirical probability $p(\sigma|s)$ that a symbol σ will be observed immediately after s is determined by the number of times σ has occurred right after s divided by the total number of times that s has occurred followed by any symbol, i.e., the empirical conditional probability of the occurrence of σ right after s in S is $p(\sigma|s) = \frac{\sum_{i=0}^{|S|-|s|} I(S_i,s\sigma)}{\sum_{i=0}^{|S|-|s|} I(S_i,s)}$.

PST is an implementation of VMM to learn significant sequence patterns. It is of compact suffix tree-structure, efficient to build, and widely used in many fields, such as gene clustering [14], musical style learning [17], outlier detection [15], text mining [16], moving object clustering [21]. The root node of the PST

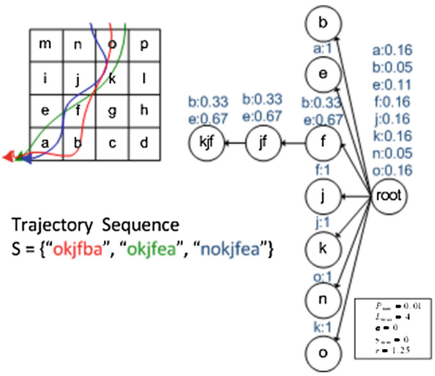


Fig. 1. Example of a trajectory sequence and the associated PST.

carries the empirical probabilities $p(\sigma), \forall \sigma \in \Sigma$ while the other nodes each carries conditional empirical probability of a significant pattern, i.e., node with label s carries the conditional empirical probability $p(\sigma|s), \forall \sigma \in \Sigma$. Each node has at most $|\Sigma|$ branches; the parent node of a node s is associated the significant pattern $suffix(s)$. The maximal depth of a PST to build is specified by L_{max}, which also determines the longest length of learned patterns. Note that every node of the PST corresponds to a significant pattern defined as below:

Definition 1 (Significant Pattern). *A subsequence $s = \sigma_0\sigma_1...\sigma_{n-1}$ is a significant pattern if*

(1) $p(s) > P_{min}$
(2) $\exists \sigma \in \Sigma$ *such that*
(a) $p(\sigma|s) \geq \gamma_{min}$
(b) $\frac{p(\sigma|s)}{p(\sigma|suffix(s))} \geq r$ *or* $\frac{p(\sigma|s)}{p(\sigma|suffix(s))} \leq \frac{1}{r}$.

The first condition with the threshold P_{min} is specified to limit the occurrence probability to find relatively important patterns. The second condition constrains the uniqueness of a pattern; it first uses γ_{min} and r to constrain that at least a relatively significant conditional probability of a pattern must be different from that of its suffix. Figure 1 shows an example of trajectory sequence and the associated PST.

2.2 Uncertain Trajectory Sequence and Redefinition of Significant Pattern

While uncertainties are considered, an uncertain trajectory sequence with length L is represented by

$$S = S_0S_1...S_{L-1} = e\begin{bmatrix} p(\sigma_0) \\ p(\sigma_1) \\ ... \\ p(\sigma_{|\Sigma|-1} \end{bmatrix}_0 \begin{bmatrix} p(\sigma_0) \\ p(\sigma_1) \\ ... \\ p(\sigma_{|\Sigma|-1} \end{bmatrix}_1 \cdots \begin{bmatrix} p(\sigma_0) \\ p(\sigma_1) \\ ... \\ p(\sigma_{|\Sigma|-1} \end{bmatrix}_{L-1},$$

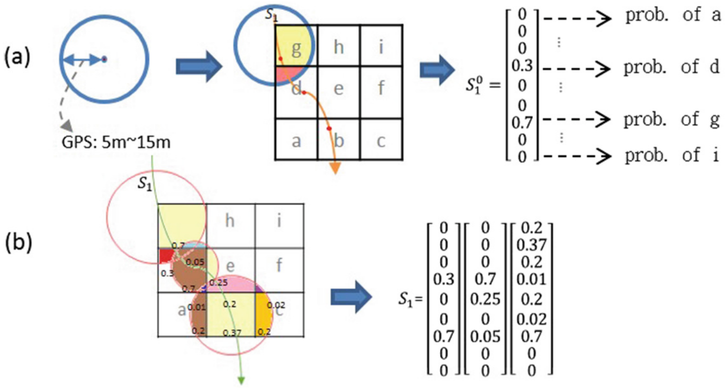


Fig. 2. Example of (a) an uncertain location and (b) an uncertain trajectory sequence.

where $p(\sigma_i)$ denotes the probability of location σ_i. For example, assuming the accuracy of GPS is with error about 5 m to 15m, an position is translated into an uncertain location, as shown in Fig. 2(a). Then, tracking an object for three time slots can get an uncertain trajectory sequence as shown in Fig. 2(b).

To mine significant patterns in uncertain trajectory sequences, we redefine $p(s)$ and $p(\sigma|s)$ of Definition 1 as $p_u(s) = \frac{\sum_{i=0}^{L-|s|} \Pi_{j=0}^{|s|-1} p_i(s_j)}{L-|s|+1}$ and $p_u(\sigma|s) = \frac{p_u(s\sigma)}{\sum_{x\in\Sigma} p_u(sx)}$. Moreover, assume existing K, l, and r such that a length L uncertain trajectory sequence can be partitioned into K segments, where the first $K-1$ partitions are with length l and the remaining one partition is with length r. Then, we prove that the computing of $p_u(s)$ can be done distributedly as below.

$$p_u(s) = \frac{\sum_{i=0}^{L-|s|} \Pi_{j=0}^{|s|-1} p_i(s_j)}{L-|s|+1} = \frac{\sum_{k=0}^{K-2} \sum_{i=0}^{l-|s|} \Pi_{j=0}^{|s|-1} p_i(s_j) + \sum_{i=0}^{r-|s|} \sum_{j=0}^{|s|-1} \Pi_{j=0}^{|s|-1} p_i(s_j)}{L-|s|+1}$$

Since $p(\sigma|s)$ is defined on $p(s)$, it is intuitive that the computing of $p(\sigma|s)$ can be done distributedly. Accordingly, based on the re-definition of $p(s)$ and $p(\sigma|s)$ and the above proof, we propose the uPST$_{MR}$ and uPST$^{+}_{MR}$ algorithms in next section.

3 The Proposed Algorithms

Under the MapReduce architecture, the functions realized by the programmer include Map and Reduce, which are automatically assigned to multiple machines by the engine of MapReduce. When the user inputs a group of Key/Value, mappers will generate a group of intermediate Key/Value while reducers will combine all related intermediate values to produce the final results.

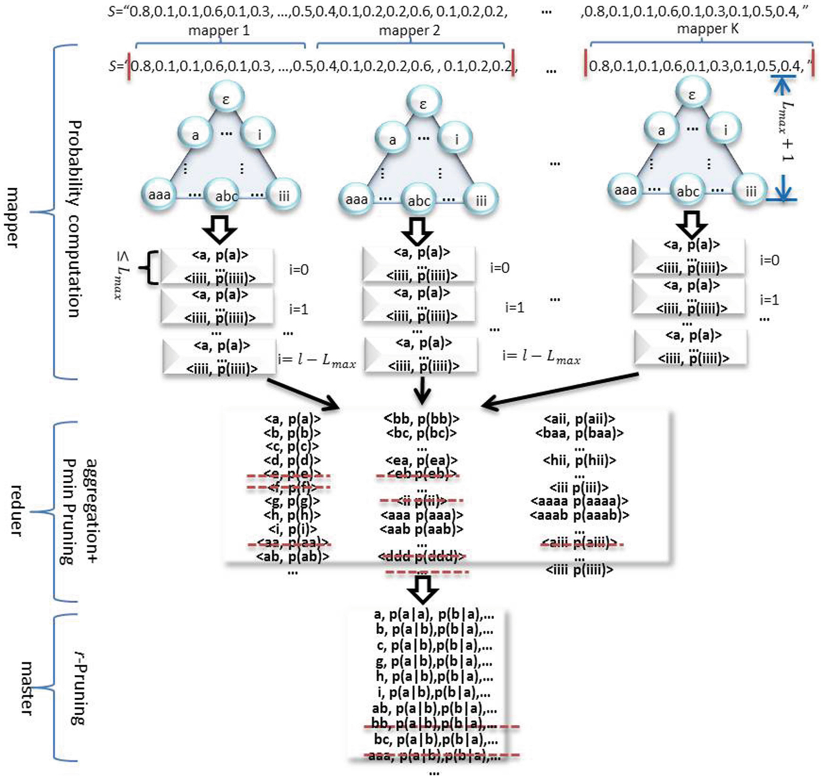


Fig. 3. Example of the naive method.

Note that the construction of a PST is mainly to calculate the conditional probability distribution of each node and exam whether it's qualified. The difficulty of constructing a PST in a distributed manner is caused by Condition 2 of Definition 1, i.e., we need to compare a child node with its parent to confirm whether it is qualified. A naive approach is to consign the mission of probability computation to MapReduce and then examine nodes' qualification at the master node. As shown in Fig. 3, the naive approach divides the input data into a few sequence sections, each of which is assigned to a mapper as input. To construct a PST with maximal depth L_{max}, each mapper computes the probabilities of subsequences with length $\leq L_{max} + 1$ and does local aggregation and the probabilities are sent to the reducer for global aggregation. Also, the reducer performs the first stage screening, i.e., pruning unqualified subsequences according to the Condition 1 of Definition 1. Then, the outputs of reducers are sent back to the master node for stage 2 screening, i.e., γ-Pruning according to Condition 2 of Definition 1. Finally, qualified subsequences together with their conditional probabilities are generated as the final results. Note that in the naive algorithm, each mapper generates a (key=subsequence, value=occurrence prob.) pair for a subsequence that ever occurs in its assigned data block. For a specified maximal tree depth (or maximal pattern length) L_{max} and Σ, the maximal number of (key, value) pairs generated is $K \times |\Sigma|^{(i+1)}$, where K denote the number of mappers.

The problems of the naive method include two folds. First, due to the distributed computing, before combining the occurrence counts of a candidate pattern from one or multiple mappers, each mapper or reducer cannot confirm whether an individual candidate pattern is qualified or not. Thus, it at most has to calculate the probabilities of all candidate patterns of a complete tree with depth L_{max}. Unfortunately, CPU resource is wasted for unqualified nodes which are in general the majority. Second, all mappers have to send a (key,value) pair for every candidate pattern to the reducers for further combination, P_{min}-Pruning and γ-Pruning. In other words, mappers cannot prune any unqualified nodes and thus the intermediate (key,value) pairs sent by the mappers could be huge. Moreover, before a reducer starts to function, all (key,value) pairs have to be sorted and merged according to their keys, which is every time consuming. Thus, when the amount of (key,value) pairs is huge, the shuffling time cost is significant. Therefore, to overcome the weaknesses of the naive approach, we propose the uPST$_{MR}$ algorithm as shown in Fig. 6 to construct the PST in a progressive, multi-layered and iterative way. Specifically, by specifying the number of layers to construct in each MapReduce iteration, we can avoid excessive mining of unqualified patterns and balance distributed computing overhead. Figure 4 shows our idea and at most $\lceil L_{max}/d \rceil$ iterations are required for constructing the whole PST. Specifically, the design has the following two main advantages: First, it decreases the number of (key, value) pairs output by mappers. Since each iteration will develop a sub-tree of d layers for each qualified leaf node that is generated in last iteration, there will be a limit amount of candidates dependent on the subtree size as well as the number of subtrees. Due to the definition of significant patterns, the number of qualified leaf nodes is generally very small. As there is an upper limit on the candidates to construct in an iteration, the shuffling time is reduced and can be tuned by varying the parameter d. Second, it accelerates γ-pruning. As the number of candidate patterns generated by a single iteration is smaller, the size of the file output by the reducer will become smaller

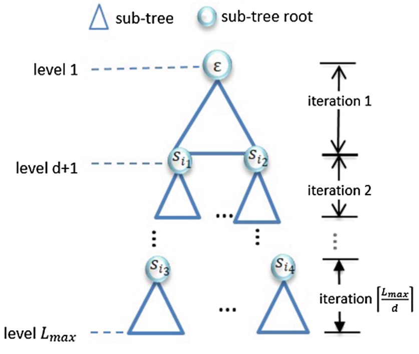


Fig. 4. The progressive, multi-layered and iterative concept of uPST$_{MR}$.

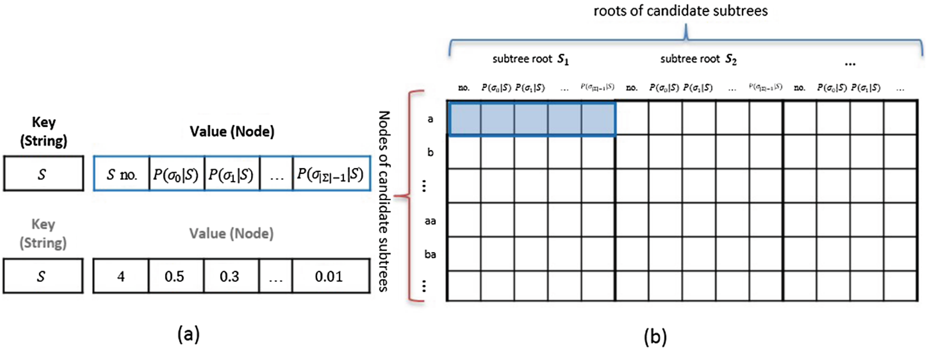
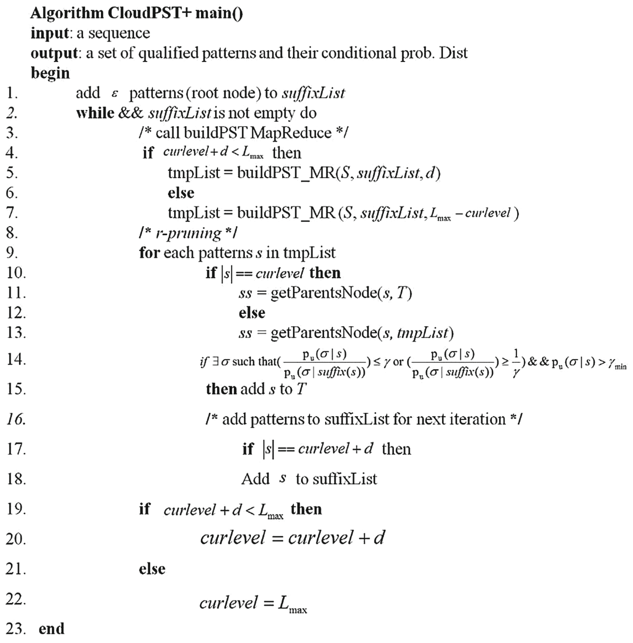


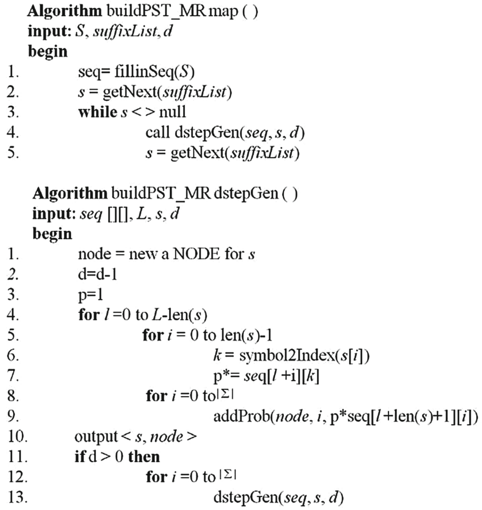
Fig. 5. Data structures of (a) (key,value) pairs and (b) NodeArray.



```
Algorithm CloudPST+ main()
input: a sequence
output: a set of qualified patterns and their conditional prob. Dist
begin
1.      add $\varepsilon$ patterns (root node) to suffixList
2.      while && suffixList is not empty do
3.            /* call buildPST MapReduce */
4.            if $curlevel + d < L_{\max}$ then
5.                tmpList = buildPST_MR(S, suffixList, d)
6.                else
7.                tmpList = buildPST_MR (S, suffixList, $L_{\max} - curlevel$)
8.            /* r-pruning */
9.            for each patterns s in tmpList
10.                 if $|s| == curlevel$ then
11.                     ss = getParentsNode(s, T)
12.                     else
13.                     ss = getParentsNode(s, tmpList)
14.                 if $\exists \sigma$ such that$(\frac{p_u(\sigma|s)}{p_u(\sigma|suffix(s))}) \le \gamma$ or $(\frac{p_u(\sigma|s)}{p_u(\sigma|suffix(s))}) \ge \frac{1}{\gamma}) \&\& p_u(\sigma|s) > \gamma_{\min}$
15.                 then add s to T
16.                 /* add patterns to suffixList for next iteration */
17.                     if $|s| == curlevel + d$ then
18.                     Add s to suffixList
19.           if $curlevel + d < L_{\max}$ then
20.                 $curlevel = curlevel + d$
21.           else
22.                 $curlevel = L_{\max}$
23.   end
```

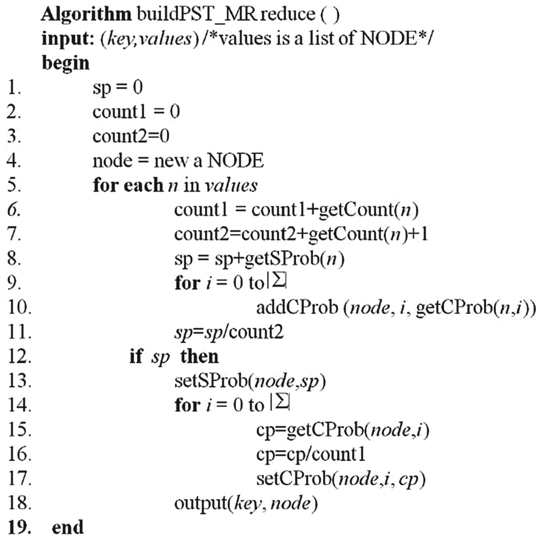
Fig. 6. The uPST$_{MR}$ algorithm.

and thus conducive to γ-prunning (condition 2) at the mater node. In view of the large amount of (key, value) pairs generated by the example of WordCount, we redesign the data structure of (key, value) pairs, as Fig. 5(a), for accumulating the statistic information for individual nodes to further lessen the number of (key, value) pairs. Figures 7 and 8 show the algorithm of the map and reduce functions of the uPST$_{MR}$ subroutine (Line 5 and Line 7 of Fig. 6) respectively. Furthermore, we propose the uPST$^{+}_{MR}$ algorithm which extend uPST$_{MR}$ with a newly designed data structure, as shown Fig. 5(b), for the temporary storage



```
Algorithm buildPST_MR map ( )
input: S, suffixList, d
begin
    seq= fillinSeq(S)
    s = getNext(suffixList)
    while s < > null
        call dstepGen(seq, s, d)
        s = getNext(suffixList)

Algorithm buildPST_MR dstepGen ( )
input: seq [][], L, s, d
begin
    node = new a NODE for s
    d=d-1
    p=1
    for l =0 to L-len(s)
        for i = 0 to len(s)-1
            k = symbol2Index(s[i])
            p*= seq[l +i][k]
        for i =0 to|Σ|
            addProb(node, i, p*seq[l +len(s)+1][i])
    output < s, node >
    if d > 0 then
        for i =0 to |Σ|
            dstepGen(seq, s, d)
```

Fig. 7. Algorithm of the map function of the uPST$_{MR}$ subroutine.



```
Algorithm buildPST_MR reduce ( )
input: (key,values) /*values is a list of NODE*/
begin
    sp = 0
    count1 = 0
    count2=0
    node = new a NODE
    for each n in values
        count1 = count1+getCount(n)
        count2=count2+getCount(n)+1
        sp = sp+getSProb(n)
        for i = 0 to|Σ|
            addCProb (node, i, getCProb(n,i))
        sp=sp/count2
      if sp then
        setSProb(node,sp)
        for i = 0 to |Σ|
            cp=getCProb(node,i)
            cp=cp/count1
            setCProb(node,i, cp)
        output(key, node)
end
```

Fig. 8. Algorithm of the reduce function of the uPST$_{MR}$ subroutine.

of the intermediate statistical results. Therefore, each iteration only needs the scanning of entire sequence for once in each iteration.

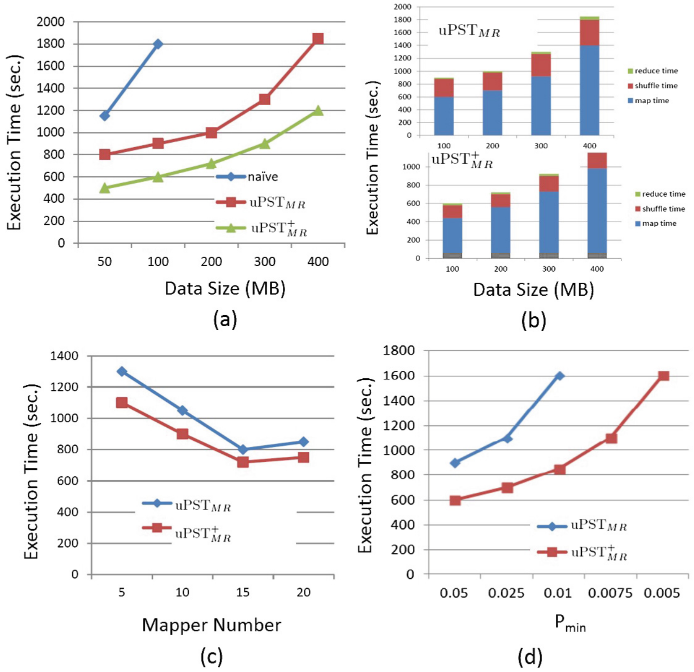


Fig. 9. Impact of data size.

4 Performance Study

For performance study, we build a small cloud environment with two PC (Intel(R) Core2 Quad CPUQ9500 @ 2.83 GHz 4 cores, 4 GB Memory) and three servers (Intel(R) Xeon CPU x3440 @ 2.53 GHz 8 cores, 8 GB Memory) with Linux Fedora14 and Hadoop-0.20.2 installed. The uncertain trajectory sequences are generated from a read dataset containing about 600 GPS trajectories and we randomly sample the generated uncertain sequences to produce the input data with required size. Figure 9(a) shows that the execution time of the naive approach dramatically augments as data size is getting larger and uPST$^+_{MR}$ outperform uPST$_{MR}$ and the naive approach. In Fig. 9(b), it can be seen that the mapping time of uPST$^+_{MR}$ is shorter than uPST$_{MR}$. In addition, not only mapping time but also shuffling time is shorten by using the OneScan approach. This is because uPST$_{MR}$ outputs a (key,value) pair for a significant pattern, or say a PST node, after a scan so that it recurrently scans and outputs (key,value) pair, which also prolongs the overall shuffling time. Figure 9(c) shows the impact of mapper number. The execution times of both algorithms first decrease inversely with the number of mappers and then they reversely increase. This is because as more mappers join in the computing, each mapper has to compute the (key,value) pairs for a complete subtree so that the overall (key,value) pairs increase; consequently, the benefit of adding more mappers is nullified and the execution time reversely increases. Finally, we study the impact of the PST size, in terms of P_{min}. Note that a small P_{min} is prone to have more patterns and a deeper tree. Figure 9(d) shows that with one-scan design, uPST$^{+}_{MR}$ is more capable of handling a large tree.

5 Conclusion

To address the problems of the ever increasing trajectory data and the mining of uncertain sequence data, this paper proposes two MapReduce algorithms, uPST$_{MR}$, and uPST$^{+}_{MR}$, to discover the patterns from a large amount of uncertain sequence data. To overcome the overhead of distributed computing, we consider both the properties of a PST and the Hadoop platform in our design. First, since longer patterns are rarely qualified, our algorithms construct a PST in a progressive, multi-layered, and iterative manner so that we can construct a fixed length of sub-trees in each iteration and delete the unqualified nodes as well as their child nodes earlier. Second, since we learn a few layers in an iteration, the number of a mapper's output (key, value) pairs is limited so that the shuffling cost is reduced. In addition, the newly designed (key, value) data structure helps reduce the shuffling cost. Third, the uPST$^{+}_{MR}$ further incorporates a two dimensional data structure, named NodeArray, to store temporary statistics data so that it can compute the probabilities of the subtrees by scanning the input data only once. To study the performance of our algorithms, we conduct several experiments. The experimental results show that uPST$^{+}_{MR}$ outperforms the naive approach significantly and show good scalability and stability. Also, although using NodeArray costs a little extra memory, the execution time is significantly lowered.

Acknowledgment. The work was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC102-2220-E-005-007.

References

1. Roux, C., Bernard, R.T.F.: Home range size, spatial distribution and habitat use of elephants in two enclosed game reserves in the eastern cape province, south Africa. Afr. J. Ecol. **47**(2), 146–153 (2007)
2. Jin, J.-F., Liu, B.-F., Yu, X., Lu, C.-H.: Wintering and migration of black-faced spoonbill in Xinghua Bay, Fujian Province. Chin. J. Zool. **44**(1), 47–53 (2009)
3. Wang, Y., Lim, E.-P., Hwang, S.-Y.: Efficient mining of group patterns from user movement data. DKE **57**(3), 240–282 (2006)
4. Li, Y., Han, J., Yang, J.: Clustering moving objects. In: ACM SIGKDD, pp. 617–622 (2004)
5. Aggarwal, C.C., Yu, P.S.: A survey of uncertain data algorithms and applications. IEEE TKDE **21**(5), 609–623 (2008)
6. Gezici, S.: A survey on wireless position estimation. Wireless Pers. Commun. **44**(3), 263–282 (2007)

7. Gu, Y., Lo, A., Niemegeers, I.: A survey of indoor positioning systems for wireless personal networks. IEEE Commun. Surv. Tutorials **11**(1), 13–32 (2009)
8. Wang, J., Luo, Y., Zhao, Y., Le, J.: A survey on privacy preserving data mining. In: 1st International Workshop on Database Technology and Applications (2009)
9. Chen, R., Fung, B.C.M., Mohammed, N., Desai, B.C., Wang, K.: Privacy-preserving trajectory data publishing by local suppression. Inf. Sci. Spec. Issue Data Min. Inf. Secur. **231**, 83–91 (2013)
10. Kuijpers, B., Othman, W.: Trajectory databases: data models, uncertainty and complete query languages. J. Comput. Syst. Sci. **76**(7), 538–560 (2009)
11. Yang, J., Hu, M.: TrajPattern: mining sequential patterns from imprecise trajectories of mobile objects. In: Ioannidis, Y., Scholl, M.H., Schmidt, J.W., Matthes, F., Hatzopoulos, M., Böhm, K., Kemper, A., Grust, T., Böhm, C. (eds.) EDBT 2006. LNCS, vol. 3896, pp. 664–681. Springer, Heidelberg (2006)
12. Leung, C.K.-S., Brajczuk, D.A.: Efficient algorithms for mining constrained frequent patterns from uncertain data. In: ACM SIGKDD Workshop on knowledge, Discovery from Uncertain Data (2009)
13. Ron, D., Singer, Y., Tishby, N.: Learning probabilistic automata with variable memory length. In: 7th Annual Conference on Computational Learning Theory, pp. 35–46 (1994)
14. Bejerano, G., Yona, G.: Variations on probabilistic suffix trees: statistical modeling and prediction of protein families. Bioinformatics **17**(1), 23–43 (2001)
15. Sun, P., Chawla, S., Arunasalam, B.: Mining for outliers in sequential databases. In: SDM (2006)
16. Pampapathi, R.M.: Annotated suffix trees for text modelling and classification. Dissertation, University of London (2008)
17. Dubnov, S., Assayag, G., Lartillot, O., Bejerano, G.: Using machine-learning methods for musical style modeling. J. Comput. **36**(10), 73–80 (2003)
18. Ghoting, A., Makarychev, K.: I/O efficient algorithms for serial and parallel suffix tree construction. ACM TODS **35**(4), 1–37 (2010)
19. Gao, F., Zaki, M.J.: Psist: a scalable approach to indexing protein structures using suffix trees. J. Parallel Distrib. Comput. **68**(1), 54–63 (2008)
20. Pellicer, S., Chen, G., Chan, K.C.C., Pan, Y.: Distributed sequence alignment applications for the public computing architecture. IEEE T-NB **7**(1), 35–43 (2008)
21. Tsai, H.-P., Yang, D.-N., Chen, M.-S.: Mining group movement patterns for tracking moving objects efficiently. IEEE TKDE **23**(2), 266–281 (2011)

Big Data Science and Engineering on E-Commerce

Two Algorithms Under Stochastic Gradient Descent Framework for Recommender Systems

Tian-Hsiang Huang[1](✉) and Vladimir Nikulin[2]

[1] Wireless Broadband Communication Protocol Cross-Campus Research Center,
National Sun Yat-sen University, Kaohsiung, Taiwan
{huangtx,vnikulin.uq}@gmail.com
[2] Department of Mathematical Methods in Economy,
Vyatka State University, Kirov, Russia

Abstract. The Recommender System (RS) is a subfield of machine learning that aims in creating algorithms to predict user preferences based on known user ratings or user behavior in selecting or purchasing items. Such a system has great importance in applications to sport, marketing and education. In the last case, we are interested to improve state of the art in student evaluation by predicting whether a student will answer the next question correctly. This prediction will help student to get right orientation which learning area should to be given greater attention. Note that the available data are given in the form of list, but not in the traditional form of matrix. Consequently, standard factorization technique is not applicable here. However, stochastic gradient methods work well with the lists of data, where the most of relations are missing, and maybe required to be predicted. In this paper we consider optimization of the most important regulation parameters, such as numbers of factors, learning and regularization rates, numbers of global iterations. Our study is based on the Grockit and Chess data, which were used online during popular data mining contests on the platform Kaggle.

Keywords: Matrix factorization · Collaborative filtering · Recommender system · Online education · Chess ratings · Pairwise coupling · Unsupervised learning

1 Introduction

The Recommender system attempts to profile user preferences over items, and models the relation between users and items. The task of recommender systems (RSs) is to recommend items that fit users tastes, in order to help the user in selecting/purchasing items from an overwhelming set of choices [1]. Such systems have great importance in applications such as e-commerce, subscription based services, information filtering, etc. Recommender systems providing personalized suggestions greatly increase the likelihood of a customer making a purchase compared to unpersonalized ones. Personalized recommendations are especially important in markets where the variety of choices is large, the taste

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 219–230, 2014.
DOI: 10.1007/978-3-319-13186-3_21

of the customer is important, and last but not least the price of the items is modest. Typical areas of such services are mostly related to art (esp. books, lectures, movies, music), education, sport, fashion, food and restaurants, gaming and humor.

The recommender system plays an important role in such highly rated Internet sites as Amazon.com, YouTube, Netflix, Yahoo, Grockit and Yelp. Moreover many media companies are now developing and deploying RSs as part of the services they provide to their subscribers. There are dedicated conferences and workshops related to the field. We refer specifically to ACM Recommender Systems (RecSys), established in 2007 and now the premier annual event in recommender technology research and applications [2,3]. At institutions of higher education around the world, undergraduate and graduate courses are now dedicated entirely to recommender systems; tutorials on RSs are very popular at computer science conferences; and recently a book introducing RSs techniques was published [4].

Matrix factorization, as a primary tool in the field of RSs, have become popular research topics due to their obvious usefulness [5]. Examples of successful matrix factorization methods are singular value decomposition and nonnegative matrix factorization (NMF) [6].

However, large applications [7], which aim at discovering and capturing the interactions between two entities, involve matrices with millions of rows (representing customers, for example, players or students) and millions of columns (representing items, for example, goods, services, opponent players or tasks), and billions of entries (relations between customers and items). In practice, most of the observations or entries are not available or missing. Accordingly, it appears to be natural to consider available data not in the form of matrix but as a list. Further, in the most of cases it is logical to employ for quality measurement loss function as a sum of terms corresponding to the particular observations. Generally, minimization of the sum with millions of terms and tens of thousands of regulation parameters is hardly possible, and the key idea of the stochastic gradient descent is to minimize particular terms sequentially one after another, where the involved parameters will be updated according to the corresponding (local) gradients. It is very essential to note that any particular observation (measurement) doesn't represent the whole list of data, and any local update should be conducted carefully with learning rates, which must be small enough to ensure stability of the learning process. Consequently, and in order to achieve high quality of approximation it will be necessary to conduct not less than 20 global iterations. During those iterations some particular parameters may experience thousands of updates, and regularization restricting growth of the parameters (according to the absolute value) is a very essential.

As an alternative to the popular factorization methods, we developed a novel and very fast algorithm for gradient-based matrix factorization (GMF), which was introduced in our previous studies [8,9], where bioinformatics was considered as an area of application. One of the subjects of this paper is a more advanced version of the GMF. We call this algorithm as GMF with two regularised learning

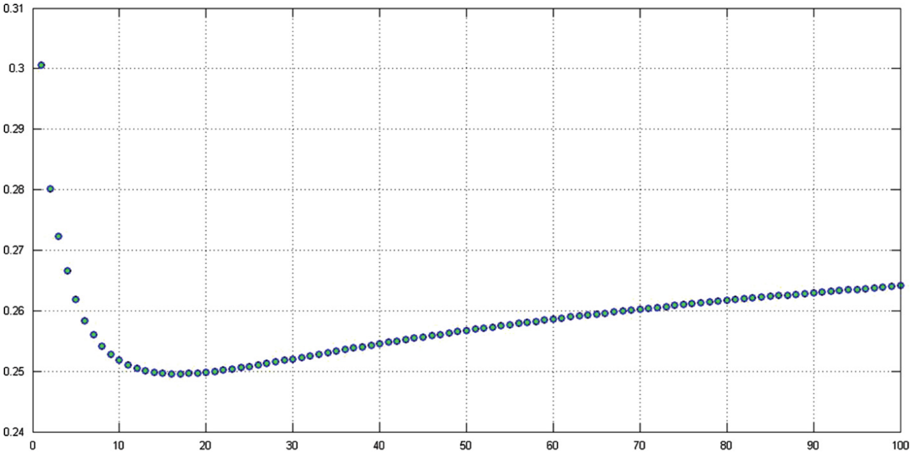


Fig. 1. Convergence of the LF + SGD algorithm: $L(\mathbf{A}, \mathbf{B}, 0, 0)$ as a function of the global iteration - horizontal axis.

rates or, simply, List Factorization under Stochastic Gradient Descent framework (LF + SGD). Details about this algorithm are given in Sect. 2. The main features of the LF + SGD are two learning rates in accordance to the number of factor matrices. By the definition, the learning process includes regularization as a very essential component.

In Sect. 3 we consider problem related to the educational area: prediction how student will complete new task based on the available historical data. As another example, where SGD is applicable, we consider historical sequence of chess games for the period of more than 10 years. In Sect. 4.1 we present an algorithm under stochastic gradient descent framework CR + SGD for calculation of the Chess Ratings (or ratings of the Chess players). We note that the main ideas behind this algorithm were motivated by [10].

2 List Factorization Under Stochastic Gradient Descent Framework (LF + SGD)

Following [11], we use the following notations in this paper. The rating matrix is denoted by $\mathbf{X} \in \{0, 1\}^{I \times J}$, where the element $x_{ij} = 1$ indicates that ith student completed jth task correctly, and incorrectly, otherwise. I and J denote the total number of students and tasks, respectively. We refer to the set of all known (i, j) pairs (test results) in $\mathbf{X}$ as $\mathcal{R}$. Superscript "hat" denotes the prediction of the given quantity: $\hat{x}$ is a prediction of x.

For recommender studies, the number I of students is typically in the hundreds of thousands, and the number J of tasks is typically in thousands. The data are represented by a rating matrix $\mathbf{X}$ of size $I \times J$, which is sparse, because the most of the elements are missing or not available. Our goal is to find a small

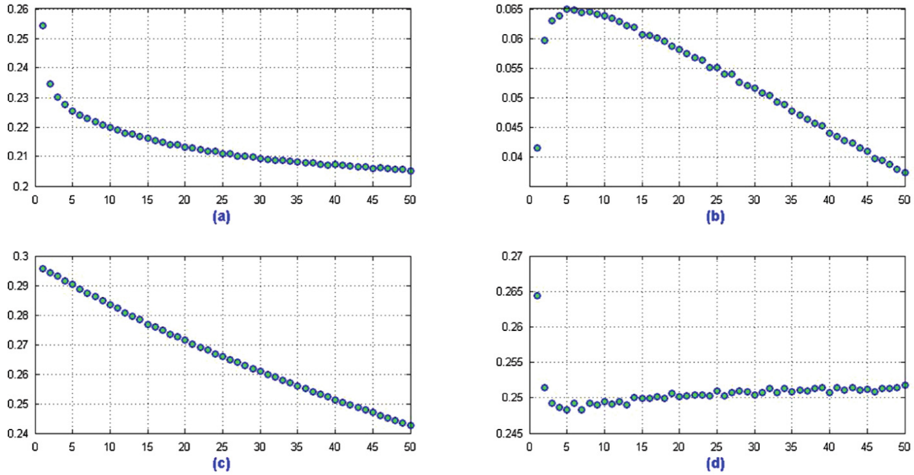


Fig. 2. Selection of the number of factors K (horizontal axis), see Sect. 3.2: (a) and (c) resubstitution (training) errors in the cases of the original and random labels; (b) the difference between (c) and (a); (d) validation error, corresponding to (a) with optimal selection: $K = 5$.

number $K \ll \min(I, J)$ of meta-students or factors. After that, we can approximate the matrix of scores as a linear combinations of those meta-students and meta-tasks. Mathematically, this corresponds to factoring matrix $\mathbf{X}$ into two matrices

$$\mathbf{X} \sim \mathbf{AB}, \tag{1}$$

where weight matrix $\mathbf{A}$ has size $I \times K$, and the factor matrix $\mathbf{B}$ has size $K \times J$, with each of K rows representing the meta-student rating pattern of the corresponding meta-task.

The main feature of the stochastic gradient descent: it doesn't require that the input data of $\mathbf{X}$ is presented in the form of matrix. In fact, the following below Algorithm 1 works not with matrix, but with list of data.

The Eq. (1) represents not an exact relation, but an approximation, where consecutive prediction for any pair (i, j) of the matrix of ratings $\mathbf{X}$ is to be computed as follows

$$\hat{x}_{ij} = \frac{1}{1 + \exp(-s_{ij})}, \tag{2}$$

where $s_{ij} = \sum_{k=1}^{K} a_{ik} b_{kj}$.

The task is to maximize the following loss function (binomial deviance)

$$L(\mathbf{A}, \mathbf{B}, \mu_1, \mu_2) = \frac{1}{\#\mathcal{R}} \sum_{(i,j) \in \mathcal{R}} H_{ij}, \tag{3}$$

$$H_{ij} = \frac{1}{\log(10)} E_{ij} + \mu_1 \sum_{k=1}^{K} a_{ik}^2 + \mu_2 \sum_{k=1}^{K} b_{kj}^2, \tag{4}$$

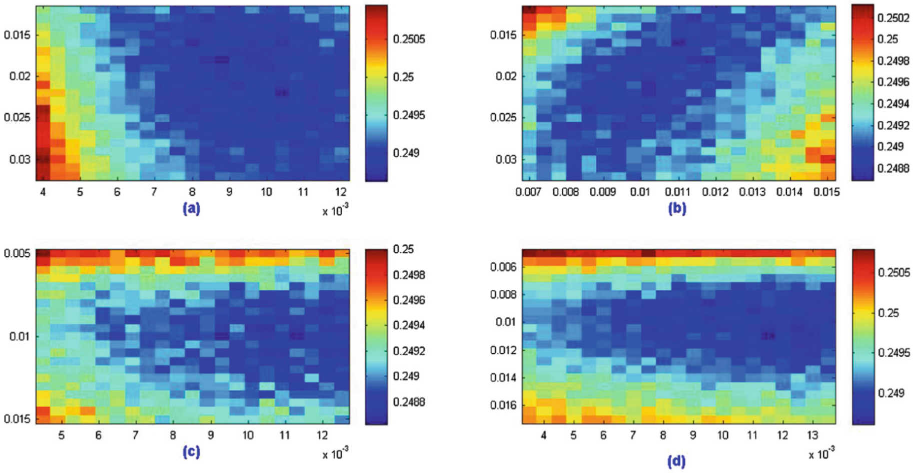


Fig. 3. Selection of the learning rates $\lambda_d \geq 0$ and regularization parameters $\mu_d \geq 0, d = 1, 2$, see Sect. 3.2.

Algorithm 1. LF + SGD with squared regularization.

```
1. Input: X - matrix (organised technically as a list) of scores, where most of the
   elements maybe missing or not available.
2. Select $N$ - number of global iterations; $K \geq 1$ - number of factors; $\lambda_d > 0, \mu_d >$
   $0, d = 1, 2,$ - learning and regularization rates (used squared regularization).
3. Initial factor matrices A and B are generated randomly.

4. Global cycle: repeat $N$ times the following steps 5 - 16:
5. external-cycle: for $(i, j) \in \mathcal{R}$ repeat steps 6 - 16:

6. compute prediction $S = \sum_{k=1}^{K} a_{ik} b_{kj}, pr = 1/(1 + \exp(-S))$;
7. compute error of prediction: $\Delta = x_{ij} - pr$;
8. internal factors-cycle: for $k = 1$ to $K$ repeat steps 10 - 16:
9. compute $\alpha = a_{ik} b_{kj}$;
10. update $a_{ik} \Leftarrow a_{ik} - \lambda_1 \cdot (\Delta \cdot b_{kj} + \mu_1 a_{ik})$   (see (4) and (7a));
11. $S \Leftarrow S - \alpha + a_{ik} b_{kj}, pr = 1/(1 + \exp(-S))$;
12. $\Delta = x_{ij} - pr$;
13. compute $\alpha = a_{ik} b_{kj}$;
14. update $b_{kj} \Leftarrow b_{kj} - \lambda_2 \cdot (\Delta \cdot a_{ik} + \mu_2 b_{kj})$   (see (4) and (7b));
15. $S \Leftarrow S - \alpha + a_{ik} b_{kj}, pr = 1/(1 + \exp(-S))$;
16. $\Delta = x_{ij} - pr$;

17. Output: A and B - matrices of latent factors.
```

where $\mu_d \geq 0, d = 1, 2$ are ridge (regularization) parameters:

$$E_{ij} = -x_{ij} \log(\hat{x}_{ij}) - (1 - x_{ij}) \log(1 - \hat{x}_{ij}), \tag{5}$$

$\log(\cdot)$ is a function of natural logarithm.

Remark 1. The regularization terms in (4) are very essential. The target of regularization is to make elements of the factor matrices as small as possible or not to increase elements of the factor matrices (by absolute value) if not necessary.

The above target function (3) includes in total $K(I + J)$ regulation parameters and may be unstable if we minimize it without taking into account the mutual dependence between elements of the matrices **A** and **B**.

In accordance with (2), we can re-write (5) in the following form

$$E_{ij} = -x_{ij} (s_{ij} - \log(1 + \exp(s_{ij}))) + (1 - x_{ij}) \log(1 + \exp(s_{ij})). \tag{6}$$

Derivatives of the function E_{ij} are given below:

$$\frac{\partial E_{ij}}{\partial a_{ik}} = (-x_{ij} + \hat{x}_{ij})\, b_{kj}, \tag{7a}$$

$$\frac{\partial E_{ij}}{\partial b_{kj}} = (-x_{ij} + \hat{x}_{ij})\, a_{ik}. \tag{7b}$$

Remark 2. Steps 10 and 14 of Algorithm 1 represent the most important update formulas. Their structures follow directly from (4), (7a) and (7b), where regularization parameters μ may differ by a constant.

3 Grockit Data Mining Competition

The International Contest Grockit was conducted on the Kaggle platform[1] from 18th November 2011 to 29th February 2012 (103 days totally). Our result was 15th out of 241 active participants. The description of the winning method is presented in [12].

Grockit[2] is the social learning company that makes products that help people learn from other people. The main objective of Grockit is to provide students with an ideal and flexible environment for learning. Grockit's success directly dependent on the student's success. Grockit is the world's fastest growing online test preparation service for students seeking to get their best potential score on a variety of standard tests, and other tests required for college admissions. Grockit is an adaptive, personalized learning program distinguished by its unique social learning features that are proven to help people learn quickly and answer more questions correctly. Study online any time of the day, from anywhere you have Internet access. Grockit is a very convenient - no boring classrooms and lectures.

[1] http://www.kaggle.com

[2] https://grockit.com/

Grockit predicts student's score based on the answers and tracks, performances and improvements, projecting accurate score improvements. Personalized and independent study is important. So is tutoring. Grockit study plans provide practice tests, personalized insight into your weak subject areas, review of your work, and the right tutor to help students to learn quickly. Grockit.TV features award-winning teachers demonstrating problem solving techniques that can be applied to develop study plan. There are hundreds of hours of Grockit.TV content. It is like Facebook, but for learning. Combining social networking with studying, Grockit encourages academic success through peer interaction.

3.1 Grockit Data

The Grockit data includes 4851475 records for training and 93100 results were requested to be predicted. The number of students $I = 179106$, and the number of tasks $J = 6046$. Therefore, the given number of records was about 0.45 % out of the theoretically full information, which will make the rating matrix complete. The data structure is very simple and includes three columns: 1) index of student, 2) index of task and 3) result (1 - correct and 0 - incorrect).

3.2 Experiments

Figure 1 illustrates convergence of Algorithm 1 as a function of the global iteration (horizontal axis). The vertical axis represents cross-validation (CV) error with $L(\mathbf{A}, \mathbf{B}, 0, 0)$, where 500000 observations were randomly excluded from training and were used for validation (30 random trials were conducted). The following parameters were used in this experiment: $K = 5, \lambda_1 = 0.022, \mu_1 = 0.006, \lambda_2 = 0.016, \mu_2 = 0.0085$.

Remark 3. The given values of the regulation parameters $\mu_d, d = 1, 2$, were used for training only. CV was conducted against (3) with no regularization: $\mu_d = 0, d = 1, 2$.

There are 54.74 % correct answers, and we generated secondary random vector of labels with the same proportion of correct answers.

Figures 2(a) and (c) illustrate convergence of Algorithm 1 in the cases of original and random labels. We can see that the error decline in the case of original labels is very fast initially and smoothed afterwards. Apparently, behaviour during an initial period should be explained by some hidden dependencies in the data. The algorithm is discovering those dependencies very quickly. In difference, error decline in the case of random labels is rather smoothed, and may be explained by overfitting: at any step we are adding to the model $I + J$ new parameters. Figure 2(b) is the most interesting and illustrates the difference between graphs of Figs. 2(c) and (a). We can see that the difference is growing to the top point. Then, it is declining. One can expect that the top point maybe used as a criterion for the selection of the number of the factors K in Algorithm 1. This hypothesis is confirmed clearly by the last graph of Fig. 2(d), which illustrates

CV error (with no regularization). Based on Figs. 2(b) and (d), we can conclude that the optimal number of factors is $K = 5$.

Further, hotmaps on Fig. 3 illustrate behavior of the CV error depending on the parameter setting, where (a) λ_1 - vertical, μ_1 - horizontal; (b) λ_1, λ_2; (c) μ_1, μ_2; (d) λ_2, μ_2. Figure 3(c) illustrates clearly importance of the regularization in order to overcome overfitting. In all experiments here we used $k = 5$ and $N = 24$.

Remark 4. Figure 3 demonstrates CV-result of below 0.249 in the case if regulation parameters were selected properly. This result appears to be very competitive taking into account results of other contestants.

4 Chess 2010 and 2011 Data Mining Contests

The Elo rating system was developed by the Hungarian physicist Arpad Elo in the 1950's and adopted by the world chess federation[3] (FIDE) in 1970. For more than four decades the FIDE Elo system has served as the primary yardstick in the world for measuring the strength of chess players. FIDE ratings are used for determining invitations to chess tournaments including the world championship cycle, calculating specific pairings in the most chess tournaments, and granting titles such as International Master or Grandmaster.

There are several alternatives to the Elo approach. Professor Mark Glickman [13] developed the Glicko and Glicko-2 systems, which extend the Elo system by introducing additional parameters to represent the reliability and volatility of player ratings. Jeff Sonas (the Organiser of the both competitions) developed Chessmetrics[4] to maximize predictive power.

The primary target of two Chess contests on the Kaggle platform in 2010 and 2011 was to discover better methodology to evaluate ratings r of the chess players. We participated and were awarded prizes in both contests: 10th out of 252 in 2010 and 4th out of 181 in 2011. The winning approach for the Chess 2010 contest, which is based on the stochastic gradient descent, is described in [10]. The duration of the Chess 2010 contest was 106 days (from 3rd August to 17th November), and the duration of the Chess 2011 contest was 86 days (from 7th February to 4th May).

The datasets for the Chess 2011 contest include real historical data provided by the FIDE. The participants train their rating systems using a dataset of over 2327683 game results with 85250 Chess players across a recent eleven year period. After that, participants use their method to predict the outcome of a further 437361 games played among those same players during the following three months.

4.1 An Algorithm to Evaluate Chess Ratings Under Stochastic Gradient Descent Framework (CR + SGD)

The algorithm presented below has several differences compared to the corresponding version of [10]. First of all, we are considering binomial deviance (but

[3] http://www.fide.com/

[4] http://www.chessmetrics.com/cm/

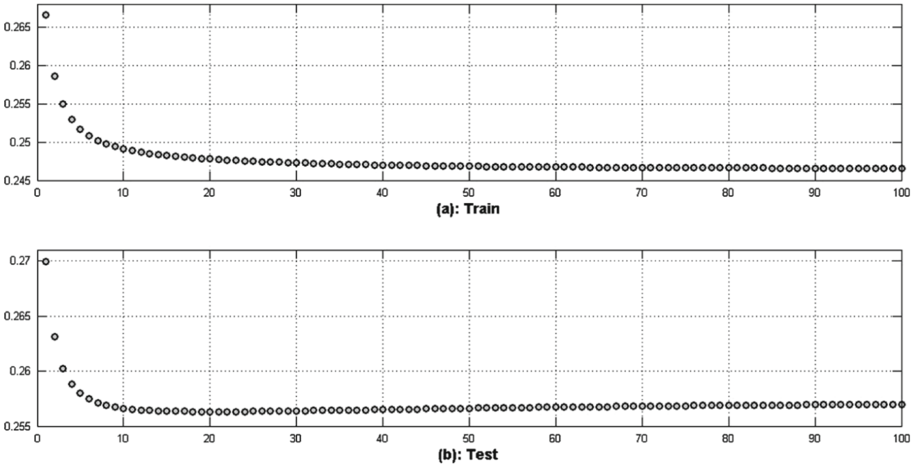


Fig. 4. Convergence of CR + SGD algorithm (see Sect. 4.1) as a function of global iteration: (a) training - is declining; (b) testing - is declining to the point $K = 21$, and growing after that.

not a squared loss), secondly, we are providing all the details regarding very essential updates of the neighbour averages (see (11)). Besides, all our constants and parameters were re-computed: optimized using special software developed in C.

According to the Bradley-Terry Model [14], the predicted probability that white player i will win against black player j is

$$\hat{p}_{ij} = \frac{1}{1 + \beta_1 \exp{(r_j - r_i)}}, \tag{8}$$

where a global parameter $0 < \gamma < 1$ reflects disadvantage of the black player (all parameters β maybe found in Table 1).

Table 1. Regulation parameters β for the CR + SGD algorithm.

β_1	β_2	β_3	β_4	β_5	β_6	β_7
0.8848	0.7626	0.1488	90.0	4.0438	1.5663	2.0317

The task is to minimize binomial deviance with squared regularization

$$\mathcal{L} = \mathcal{F} + \mathcal{R}, \ \mathcal{F} = -\frac{\alpha}{\#\mathcal{T}} \sum_{(i,j) \in \mathcal{T}} \mathcal{S}_{ij}, \tag{9}$$

where

$$\mathcal{S}_{ij} = p_{ij} \log{(\hat{p}_{ij})} + (1 - p_{ij}) \log{(1 - \hat{p}_{ij})}, \tag{10}$$

$$\mathcal{R} = \mu \sum_{i \in \mathcal{I}} (r_i - q_i)^2, \; q_i = \frac{\sum_{j \in \mathcal{N}_i} w_{ij} r_j}{\sum_{j \in \mathcal{N}_i} w_{ij}}, \tag{11}$$

$$w_{ij} = \left(\frac{1 + t_{ij} - t_{\min}}{1 + t_{\max} - t_{\min}} \right)^{\beta_2}, \tag{12}$$

where $\alpha > 0$ and $\mu > 0$ are regulation parameters, $\mathcal{N}_i$ is the list of neighbours or list of players who played with player i in the past. This list includes 1) result, 2) color, 3) month. We note the possibility of several games corresponding to the same two players in the past. Accordingly, and in order to simplify notations, we shall understand under t_{ij} (month of the game) not unique value: $1 = t_{\min} \leq t_{ij} \leq t_{\max} = 132$.

Remark 5. We note that actual values of the parameters α and μ are not important in (9), (10) and (11), which define the structure of the update formulas (14a) and (14b), where all the necessary parameters β are given in Table 1.

As a consequence of (8) and (10), the following relations are valid

$$\frac{\partial \mathcal{S}_{ij}}{\partial r_i} = \hat{p}_{ij} - p_{ij}, \tag{13a}$$

$$\frac{\partial \mathcal{S}_{ij}}{\partial r_j} = p_{ij} - \hat{p}_{ij}. \tag{13b}$$

Using (13a) and (13b), we shall derive the main update formulas

$$r_i \Leftarrow r_i - \beta_3 \lambda_k \left(w_{ij}(\hat{p}_{ij} - p_{ij}) + \beta_6 \frac{r_i - q_i}{s_i + \beta_7} \right), \tag{14a}$$

$$r_j \Leftarrow r_j - \beta_3 \lambda_k \left(-w_{ij}(\hat{p}_{ij} - p_{ij}) + \beta_6 \frac{r_j - q_j}{s_j + \beta_7} \right), \tag{14b}$$

where s_i is the number of games, corresponding to the player i during the whole period of 132 months,

$$\lambda_n = \left(\frac{1 + \beta_4}{n + \beta_4} \right)^{\beta_5},$$

and n is the index of global iteration.

Figure 4 illustrates convergence of the algorithm CR + SGD. The resubstitution error (a) is steady declining (as expected). In difference, the test error (b) is declining to some point $n = 21$. After that, it is slowly growing. Therefore, the algorithm must be stopped after 21 global iterations. There is a clear similarity between Figs. 1 and 4(b).

Remark 6. The neighbour centroids q_i (11) represent a very important element of the algorithm CR + SGD, and they must be recomputed with updated ratings after any global iteration.

Remark 7. Using algorithm CR + SGD, we can compute ratings of the Chess players according to the past 130, 131 and 132 previous months. Obviously, those ratings will be different, and we can use them for generation of the secondary features, such as derivatives and second derivatives. Combined with some other secondary features, we can create a database for R, where we can apply such functions as GBM and randomForest to compute the required probabilities of the outcomes of the games.

5 Concluding Remarks

In fact, the proposed algorithms LF + SGD and CR + SGD represent a flexible and simple structures, where we can apply and test any regulation parameters. The algorithms were written in C-code and are very fast. As a consequence, we conducted many tests (using Grockit and Chess data as examples) in order to optimize regulation parameters. In the case of LF + SGD, we have found that it is better to separate learning rates, corresponding to different factor matrices. Initially, learning rates must be big enough. Then, they should decline (see Sect. 4.1), and it is a subject of further investigation. Regularization represents an essential and a very important part of the SGD system. The number of factors must not be too big in order to prevent overfitting. In general terms, stochastic gradient descent appears to be an ideal approach applied to very large lists of data (case of sparse matrices, where only a few percents of data are available).

Finally, our method had been proven to be efficient online during popular data mining Contests on the platform Kaggle. Further improvement maybe achieved with homogeneous ensembling as discussed in [15].

Acknowledgments. The research work of Tian-Hsiang Huang was supported by the NSC 102-2410-H-110-083- project from National Science Council, Taiwan, and 102-EC-17-A-03-S1-214 project from the Ministry of Economic Affairs (MOEA) of Taiwan.

References

1. Takacs, G., Pilaszy, I., Nemeth, B., Tikk, D.: Scalable collaborative filtering approaches for large recommender systems. J. Mach. Learn. Res. **10**, 623–656 (2009)
2. Ricci, F., Rokach, L., Shapira, B., Kantor, P.: Recommender Systems Handbook. Springer, Heidelberg (2011)
3. Zhuang, Y., Chin, W.S., Juan, Y.C., Lin, C.J.: A fast parallel SGD for matrix factorization in shared memory systems. In: RecSys 2013, October 12–16, 2013, Hong Kong, China (2013)
4. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender Systems an Introduction. Cambridge University Press, New York (2010)
5. Oja, E., Ilin, A., Luttinen, J., Yang, Z.: Linear expansions with nonlinear cost functions: modelling, representation, and partitioning. In: Aranda, J., Xambo, S. (eds.) Plenary and Invited Lectures, WCCI 2010, Barcelona, Spain, pp. 105–123 (2010)

6. Lee, D., Seung, H.: Learning the parts of objects by nonnegative matrix factorization. Nat. **401**, 788–791 (1999)
7. Gemulla, R., Haas, P., Nijkamp, E., Sismanis, Y.: Large-scale matrix factorization with distributed stochastic gradient descent. In: KDD 2011 Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 69–77 (2011)
8. Nikulin, V., McLachlan, G.: Classification of imbalanced marketing data with balanced random sets. JMLR Workshop Conf. Proc. **7**, 89–100 (2009)
9. Nikulin, V., Huang, T.H.: Unsupervised dimensionality reduction via gradient-based matrix factorization with two learning rates and their automatic updates. J. Mach. Learn. Res. Workshop Conf. Proc. **27**, 181–195 (2012)
10. Sismanis, Y.: How i won the chess ratings - Elo vs the rest of the world competition. In: arXiv:1012.4571v1 (2010)
11. Takacs, G., Pilaszy, I., Nemeth, B., Tikk, D.: On the gravity recommendation system. In: KDD Cup Workshop at SIGKDD 2007 San Jose, California USA, pp. 22–30 (2007)
12. Rendle, S.: Factorization machines with libFM. ACM Trans. Intell. Syst. Technol. **3**, 1–22 (2012)
13. Glickman, M.E.: Parameter estimation in large dynamic paired comparison experiments. Appl. Statist. **48**(3), 377–394 (1999)
14. Huang, T.K., Weng, R., Lin, C.J.: Generalized Bradley-Terry models and multi-class probability estimates. J. Mach. Learn. Res. **7**, 85–115 (2006)
15. Nikulin, V., Bakharia, A., Huang, T.-H.: On the evaluation of the homogeneous ensembles with CV-passports. In: Li, J., Cao, L., Wang, C., Tan, K.C., Liu, B., Pei, J., Tseng, V.S. (eds.) PAKDD 2013 Workshops. LNCS, vol. 7867, pp. 109–120. Springer, Heidelberg (2013)

Scalable Textual Similarity Search on Large Document Collections Through Random Indexing and K-means Clustering

Ali Cevahir(✉)

Rakuten Institute of Technology, Rakuten Inc., Tokyo, Japan
ali.cevahir@mail.rakuten.com

Abstract. This work presents an index partitioning technique for large-scale text-based search engines. Large e-commerce sites contain millions of products visited by millions of users. Textual similarity search has many uses in e-commerce sites, for instance in building recommendation engines. However, the size of the corpus makes it prohibitive to use naive approaches for real-time search. In order to reduce response times, the search is executed within a small subset of most related documents. To achieve this goal, documents are clustered using k-means. However, vectors used for k-means clustering are very high dimensional. Random indexing is applied to reduce dimensionality. We boosted these steps with GPUs to reduce preprocessing overheads. Once clusters are built, text queries are executed within the closest clusters. Our experiments on a large document collection for a recommendation scenario reveal that only 1.7 % loss in recommendation precision is possible by realizing only 28 % of search operations in the inverted index.

Keywords: Textual similarity search · Vector-space model · Inverted index · Random indexing · K-means

1 Introduction

Textual similarity search is one of the key components of the e-commerce sites. It is not only used in the product search engine, but used in many different components, such as in analytics engines or content-based recommendation systems. Today's large e-commerce sites need to be capable of handling thousands of queries per second amongst millions of products. For example, there are around 89 million registered users and more than 150 million products are being listed in Rakuten Ichiba[1] which is the largest e-commerce site in Japan. Millions of these products are updated, added or removed every day.

In this work, we propose a scalable solution for textual similarity search in large document collections. We store the data in inverted index structure to be able to make similarity searches efficiently. Partitioning of the inverted index is required to scale for larger datasets. For distributed frameworks, larger number of processors is employed for larger index sizes and the index is partitioned and/or replicated through processors.

[1] http://www.rakuten.co.jp/

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 231–238, 2014.
DOI: 10.1007/978-3-319-13186-3_22

It is also possible to improve scalability without increasing processing power. To be able to do so, similarity searches are executed only within the closest partition(s), at the risk of possible loss of precision. The inverted index is partitioned through clustering of documents, so that related documents are gathered in partitions. In the vector-space model [1] for textual search, each document is represented as a high-dimensional sparse vector. It is time-consuming to cluster high dimensional vectors with conventional clustering algorithms for vector quantization. We utilize random indexing [2] for dimensionality reduction for vectors representing documents. K-means is employed to cluster vectors with reduced dimensions. Documents in each cluster are mapped to the same partition for building distributed inverted index.

We have tested our technique on a dataset containing 11.5 million product names to observe top *n* recalls and reductions in number of search operations, for different parameters. We also provide precision results for a content-based recommendation scenario, where textual search is the core component of the recommendation engine. Experimental results validate that it is possible to reduce number of search operations by 3.5 times with only 1.7 % reduction in recommendation precision.

2 Background

Each document and text query is represented as a sparse vector in the vector-space model for textual similarity search [1]. Accordingly, the document collection can be represented as a sparse matrix M which contains one row for each document and one column for each individual term. M_{ij} is nonzero if document i contains the term j. Otherwise, $M_{ij} = 0$. Nonzero values represent the weights of the terms in their documents. Term frequency – inverted document frequency is one of the most widely used measures for weighting. Textual similarity search is therefore achieved by finding the closest rows of the document matrix to the query vector.

Inverted index is a well-known data structure used for retrieving similar documents efficiently. It is a collection of lists, where each list is a mapping from a term to the documents containing that term. State of the art search engines are built based on the inverted index data structure. It is worth noting that the nature of queries for similar document search is different than typical queries made by users to full-text search engines for keyword search. Queries submitted to full-text search engines are usually "and" queries and consist of a few keywords. On the other hand, queries for document similarity search are "or" queries and may contain more keywords than that of keyword search. "Or" queries are slower than "and" queries. Therefore, efforts on scalable implementations become more significant for document similarity search.

Inverted index partitioning is studied in the context of Web search engines in [3]. *Term-based partitioning* and *document-based partitioning* are two main types of inverted index partitioning. In term-based partitioning, each partition contains disjoint subsets of terms and their associated lists of documents. In document-based partitioning, documents are distributed among partitions, and inverted index is built for each partition.

In a more related work by Bhagwat et al. [4], an inverted index partitioning method based on hash functions for similarity searches is explained. Although both works [3, 4] discuss index partitioning, their goals and target applications are completely different. In [3], authors focus on improving query retrieval times in parallel systems without changing query results. On the other hand, the goal in [4] is to partition the index cleverly so that queries return most of the similar documents when searches are executed only inside a small portion of the partitions. They report 73 % recall in top-20 similar documents when the overall recall is 24 %.

Matrix factorization is studied extensively to reduce matrix dimensions for recommender systems research. There are studies based on techniques like SVD [5], PCA [6] and maximum margin matrix factorization [7]. However, these techniques are computationally expensive, hence not feasible to be adopted for dimensionality reduction of large document matrices in text-based search engines. Also, the matrix needs to be recomputed whenever it is updated if factorization techniques are used, which is not suitable for dynamic document collections such as e-commerce data.

On the other hand, random indexing [2] is a simple yet efficient method dimensionality reduction which is based on a probabilistic model [8]. High dimensional vectors are projected to lower dimensional space by this technique while keeping the distance between vectors with high probability. It is also appropriate to be utilized in dynamic databases. Only the concerning vector is recomputed when a document is updated.

3 Inverted Index Partitioning and Search

Using the random indexing method, $d \times t$ dimensional document matrix M is mapped into $d \times k$ dimensional matrix, where d is the number of documents, t is the number of individual terms and $k << t$ is the dimension of the projected space. This is realized by multiplying M with random index matrix R as follows:

$$M'_{d\times k} = M_{d\times t} \times R_{t\times k} \tag{1}$$

The random index matrix R contains equal number of randomly assigned 1 s and -1 s in their rows. This corresponds to having an index vector for each term. Index vectors are nearly orthogonal to each other. Context vector M'_{i*} for each document i is computed by adding index vectors of its terms multiplied by the terms score in i. We use tf-idf scores for weighting terms in documents.

It is shown that it is possible to have accurate projections with sparse random indexing with R containing a few nonzero entries. Achlioptas [9] proves that having only 1/3 of the entries in R set to 1 or -1 achieves the same accuracy as dense random indexing. In [10], it is shown that very sparse random indexing by having only a few nonzeros for index vectors yields very little loss in accuracy.

Once context vectors are computed for each document as explained above, they are clustered using k-means. Document-based partitioning of the inverted index is realized by assigning documents whose context vectors are in the same cluster to the same partition. A search for a query document containing a set of terms is executed in the closest partition(s) of the distributed index, instead of the full search in the unpartitioned index. The closest partitions are found by comparing context vector computed for the query with the centers of the corresponding clusters for the partitions. Similar documents are retrieved by iterating lists of the partitioned index, instead of a vector search amongst context vectors of the concerning documents.

E-commerce data is dynamic. For some online stores, like auction sites, the change in the product database can be drastic. The proposed system in this paper handles updates quickly, without need for re-clustering for each update. New products are added to the partitions with the closest corresponding cluster centers. Inverted index entries are immediately removed for deleted items. Updating already existed item requires one deletion and on addition operation. However, as the data change considerably much, cluster centers are moved and partitioning is distorted. Re-clustering for regular intervals might be required in this case.

4 Experimental Results

In the experiments, we used a product dataset with 11,540,956 items and their titles taken from Rakuten online shops which sell books, CD-DVD and PC games. There are 860,219 words extracted from titles. Stop words were discarded. As a result, the document matrix has a size of 11,5 M × 860 K. Average title length is 6.4.

We test a content-based recommendation scenario, in which products with similar titles to the product in the active user's profile are recommended. Similar products are retrieved by searching the title of the item in the active user's profile from the closest distributed inverted indexes, as explained in the previous section.

We used 3-month purchase history as a reference for recommendation accuracy. 359,875 users who had at least 4 purchases from the related shops were selected for tests. One item from each test user was selected as the active item.

We present average recall results for top n similar documents as well as relative recommendation precisions for our technique with respect to the original unpartitioned search results. Results are presented for varying parameters of $k = \{128, 1024\}$, $n = \{5, 10, 20\}$, $C = \{64, 128, 1024\}$ and $m = \{1, 2, 3, 4, 5\}$; where k is the dimension of the context vector, n is the number of the most similar documents retrieved, C is the number of partitions (clusters) and m is the number of closest partitions to be searched. Index vectors contain ten 1 s and ten -1 s. The recall for a top n search from m indexes of the C-way partitioned index by clustering of k dimensional context vector is defined as

$$\text{Recall} = \frac{|T_n \cap T_n^{k,m,C}|}{|T_n|}, \tag{2}$$

where T_n is the set of top n result in unpartitioned index, and $T_n^{k,m,C}$ is the set of top n results from the test instance with parameters k, m and C. Hence, $|T_n \cap T_n^{k,m,C}|$ formulates the number of retrieved items from partitioned index that also exist in the search from unpartitioned index. $|T_n| = n$, if n similar items are found by the similarity search in the unpartitioned index. Average recall values are presented in Fig. 1.

Figure 2 depicts the scaled average recommendation precisions for the same parameters with Fig. 1 for n, k, m and C. Precision of a recommendation is calculated

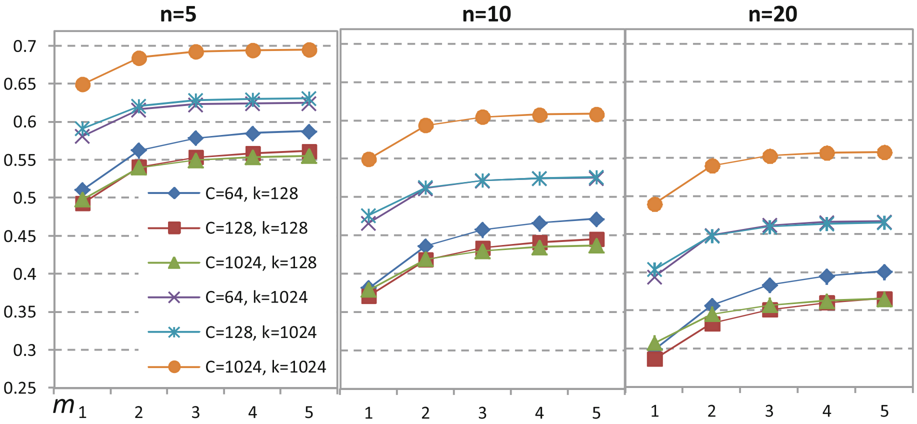


Fig. 1. Average recalls for top 5, 10 and 20 documents for varying *m*, *k* and *C* values

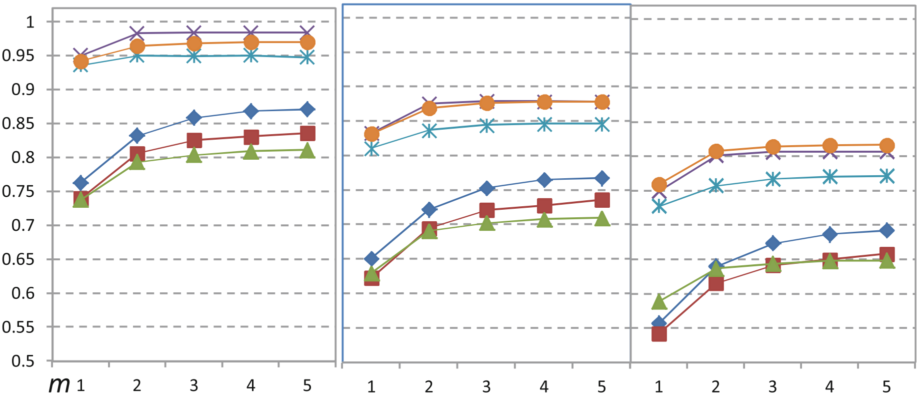


Fig. 2. Scaled average precisions for content-based recommendation for 5, 10 and 20 recommendations and varying *m*, *k* and *C* values

by comparing purchase history of the test user with recommended *n* items. It can be written as

$$\text{Precision} = \frac{|R \cap H|}{|R|}, \tag{3}$$

where *R* is the set of *n* recommendations as a result of similarity search and *H* is the history of the test user. In Fig. 2, results are scaled with the recommendation precision of the unpartitioned index (precision values are divided by the unpartitioned index search precisions) to better observe the relative change in precisions.

From the above figures, it can be seen that there is a considerable accuracy loss when *k* is chosen as 128, instead of 1024. It is not always correct that accuracy drops as the clustering factor C increases. 1024-way clustering with $k = 1024$ is the clear winner for recall values. This finding suggests that higher clustering factors can sometimes help eliminating noise occurring with lower clustering factors. It is also interesting to observe that better recalls for top n items does not necessarily mean that recommendation precisions are also better. Although top 5 recalls is the best with $k = 1024$ and $C = 1024$, precisions for 5 recommendations is better for $k = 1024$ and $C = 64$.

Number of the closest partitions searched m is another factor affecting retrieval accuracy. However, recall and precision values does not change much for $m > 2$. Accuracy drops when number of retrieved items n increase, as less-related items are distributed into different partitions.

Although there are N/C items, where N is the total number of items, on average in a partition as a result of C-way clustering, the reduction in the processing times is expected to be less than C times for search executed in one partition. It is because the search time in the inverted index is proportional to the number of entries traversed in related lists, not the number of items in the partition. We do not expect the number of traversed list entries is reduced C times for the index to be searched, as the related items to the query are gathered in the closest partition. Another reason for lower speedups is the imbalance between clusters. Nevertheless, we still observe maximum 18 times reductions in number of search operations. See Fig. 3 for average reductions in the number of traversed list entries of the inverted indexes for varying C, k and m values. As can be seen from the figure, reductions are higher for $k = 128$. This is because clusters generated by k-means are balanced better for k = 128, although recalls and precisions are lower in this case. From Figs. 2 and 3, we can conclude that recommendation precision drops only by 1.7 %, while only 28 % of the search operations are executed ($n = 5$, $C = 64$, $k = 1024$, $m = 2$).

Distance metric to compare vectors is another factor for imbalance between clusters. Using L1 distance to compare vectors during k-means clustering results with highly-imbalanced clusters. Therefore, we have used L2 distance to cluster context vectors in this paper.

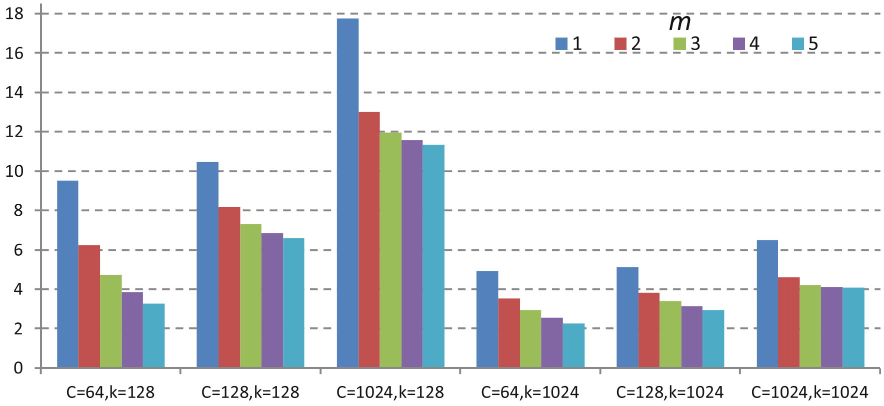


Fig. 3. Reductions in average number of traversed inverted index entries

In order to find the closest partition to be searched, vector comparisons are required, which is a drawback of the proposed technique. Vector operations might ruin speedups gained by reduction of search space for high number of partitions and relatively less number of documents to be searched. In such cases, we suggest using hardware accelerators, such as GPUs for vector comparisons. GPUs are excellent devices for vector operations. We use GPUs for the vector operations in the preprocessing step, too. For instance, k-means is very expensive for high-dimensional vectors with large number of vectors. But we achieve more than two order of magnitude speedups for k-means using GPUs, which enables us processing higher dimensional context vectors for better accuracy. Refer to [11] for implementation details of k-means on GPUs for large number of high-dimensional vectors.

5 Conclusion

In this work, we proposed a method for faster textual similarity search based on random indexing and k-means. The proposed method gathers similar documents in partitions, so that the loss in retrieval accuracy is reduced. We targeted particularly to large-scale problems requiring real-time recommendations in a dynamic environment, where the items are constantly added, removed or updated. We experimented the proposed method on a large document collection with different parameters to observe retrieval quality and speedups.

Search engine optimization is very sensitive in e-commerce websites. We do not expect this method to replace the text search engines of e-commerce sites, as it returns approximate search results. But it might be utilized for developing different components of the search engine. Similarly, lower recommendation precision might not always be tolerable. However, the method explained in the paper can be useful in many components of the e-commerce sites, such as in the analytics engines.

Computational imbalance of the inverted index partitions due to the imbalanced k-means output is a serious problem for achieving further reductions in computational cost. We leave this problem as a future work.

References

1. Salton, G., Anita, W., Chung-Shu, Y.: A vector space model for automatic indexing. Commun. ACM **18**(11), 613–620 (1975)
2. Kanerva, P., Jan K., Anders H.: Random indexing of text samples for latent semantic analysis. In: Proceedings of the 22nd Annual Conference of the Cognitive Science Society, p. 1036 (2000)
3. Cambazoglu, B.B., Catal, A., Aykanat, C.: Effect of inverted index partitioning schemes on performance of query processing in parallel text retrieval systems. In: Levi, A., Savaş, E., Yenigün, H., Balcısoy, S., Saygın, Y. (eds.) ISCIS 2006. LNCS, vol. 4263, pp. 717–725. Springer, Heidelberg (2006)

4. Bhagwat, D., Eshghi, K., Mehra, P.: Content-based document routing and index partitioning for scalable similarity-based searches in a large corpus. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07), pp. 105–112. ACM, New York (2007)
5. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
6. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: a constant time collaborative filtering algorithm. Inf. Retrieval **4**(2), 133–151 (2001)
7. Rennie, J.D., Srebro, N.: Fast maximum margin matrix factorization for collaborative prediction. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 713–719. ACM, New York (2005)
8. Johnson, W.B., Lindenstrauss, J.: Extensions of Lipschitz mappings into a Hilbert space. Contemp. Math. **26**, 189–206 (1984)
9. Achlioptas, D.: Database-friendly random projections: johnson-lindenstrauss with binary coins. J. Comput. Syst. Sci. **66**(4), 671–687 (2003)
10. Li, P., Hastie, T.J., Church, K.W.: Very sparse random projections. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06), pp. 287–296. ACM, New York (2006)
11. Cevahir, A., Torii, J.: High performance online image search with GPUs on large image databases. Int. J. Multimedia Data Eng. Manage. (IJMDEM) **4**(3), 24–41 (2013)

Complementary Product Selection in E-Commerce

Yian Chen[1](✉), Yuh-Ming Chiu[1], and Song Han[2]

[1] Yahoo EC Central, Taipei, Taiwan
{cattail,yuhming,songhan}@yahoo-inc.com
[2] Yahoo Labs, Beijing, China

Abstract. Item recommendation is considered as an important feature for e-commerce sites. Item recommendation can be categorized into alternative recommendation and complementary recommendation. Alternative item recommendation technologies are quite mature and widely adopted. However, complementary item recommendation is rarely explored although most people consider this type of recommendation very important. To the best of our knowledge, this work is the pioneer in the area of complementary recommendation. Our prototype yields very high item coverage so we can generate recommendations for most of our products. Further, our system also yields fairly good precision, i.e. items recommended are deemed relevent by editors.

1 Introduction

Providing personalized and highly relevant product recommendations is often considered as a major approach to increase the quality of the customer experience and associated customer loyalty. Product recommendations come in at least two flavors in general [1], namely, alternative and complementary. Recommendations for alternative products offer suggestions for products that are similar to the customer's choice and could be bought as an alternative or replacement. Recommendations for complementary products offer suggestions for products that would enhance, complement, complete, or go well with the selection of the users.

Alternative item recommendation technologies are quite mature and is widely used in major e-commerce sites nowadays. By offering alternative items to the users, users can either buy the item they originally searched for, buy its alternative, or buy similar items in a batch. Intuitively, people don't buy items in batches under most circumstances. Hence, the benefit of offering alternative recommendation is to increase the probability that a user buys at least one item, either the one they originally interested or one alternative from our site. In other words, we utilize alternative recommendations to improve the likelihood of at least one successful transaction, hence securing the base revenue.

Given that a user already has the intention to buy items from our site, complementary recommendation is an approach to maximize the amount of transaction. For most items, people tend to buy products in bundles as human

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 239–246, 2014.
DOI: 10.1007/978-3-319-13186-3_23

nature. For example, it is likely that a person will by a cellphone protective case while make a cellphone purchase. It is also common that a person buys apparel accessories, e.g. ties, brooches, or neckless, when making a suit/dress purchase. Complementary recommendation satisfies this need and try to encourage people making one-stop shopping hence maximize the total amount of each transaction. We can maximize the revenue of our site by combining the benefits of both alternative and complementary recommendations.

Although complementary recommendation is important, the related technologies are relatively less mature. There are literatures discussing the importance of complementary recommendations [2,3]. The actual implementation and deployment of such technologies are few and far between [4]. To the best of knowledge, this is a pioneer work in the recommendation area. This paper is organized as follows: We present two algorithms generating complementary item relationships is presented in details in Sect. 2. We present our experiment setup and demonstrate the performance of our algorithms in Sect. 3. We conclude our work and identify future direction in Sects. 4 and 5.

2 Algorithms

In our research, we propose two methods to find complementary products with high relevancy. The former is a popularity/co-occurance based filtering technique provided that we have hand-selected possible complementary product categories. We provide more details of this algorithm in Sect. 2.1. On the other hand, An algorithm involving mining complementary groups automatically using product descriptions is presented in Sect. 2.2.

2.1 Popularity/Co-occurance Filtering

Product catalog tree is a hierarchical structure that describes the relationship between different items groups. Items belonging to the same node in the product catalog tree often are listed together on the E-Commerce site. The characteristic of this tree is the most products belonging to the same node will be similar in terms of their functionalities and the style. But, on the other hand, there could be multiple nodes containing similar products.

Editors can leverage this tree to specify pairs of nodes in which the products are complementary. For example, we have the catalog tree nodes A and B representing dresses and shoes, respectively. Node A and B could be considered as complementary nodes. Items under the complementary nodes are candidates of the complementary products. Note that it is not sufficient to enumerate all items in node A along with node B as complementary product since their styles could not match well. To solve this problem, we build a model, view-also-view (VV), to help us filter those candidate with matching styles. Based on our observation, view-also-view product pairs are very consistent in terms of their styles, since they are generated by the users and the users preferences of styles may not change abruptly in short time.

VV Pairs. Given a product A, we find the products $\{B,\ C\}$ are viewed by certain fraction of people who viewed also A. By doing so, we can generate pairs of items $\{A,\ B\}$, $\{A,\ C\}$ being viewed together. We denote those pairs of products VV pairs hereafter. Similarly, we can find product pairs bought together and denote them as BB pairs.

The algorithm of finding VV pairs can be briefly described as follows. First, for a pair of product A and B, we derive $\mathbb{P}(A), \mathbb{P}(B)$, and $\mathbb{P}(A, B)$, which stand for the probabilities of A being viewed, B being viewed, and A and B being viewed together, respectively. Afterwards, we use a function F which takes $\mathbb{P}(A), \mathbb{P}(B)$ and $\mathbb{P}(A, B)$ as arguments to derive a value W represents the correlation between A and B. The higher W is, the more likely people viewed A also would like to view item B [5].

2.2 Topic Mining

The previous method is not scalable since the editors of the e-commerce sites have to manually specify complementary product catalog tree nodes. And since similar items could be contained in different sub trees, this task could be tedious. Another approach is to find complementary catalog tree nodes automatically. Our proposed method is to utilize the buy also buy product pairs to discover complementary nodes, since items bought together are very likely to be complementary with each other. However, buying events occur must less often than viewing events, using item bought together to find complementary items directly could lead to low coverage in our recommender system. We leverage buying information to find the complementary nodes instead.

Despite complementary items, buy also buy product pairs also could be constituted by two similar items if the products are consumptive. To find complementary product nodes, we first have to eliminate the buy also buy product pairs with high product similarity and those pairs constituted by items having seldom relations.

2.3 Product Similarity

To filter out similar products, we have to calculate inter-similarity between products. The similarity between a pair of items can be derived from their product information. In this research, we treat each product information as a short document. PLSI is applied to project products into a topic.

PLSI. Probabilistic Latent Semantic Indexing (PLSI) was first proposed by T. Hofmann in 1999 [6]. It is a statistical topic model for the analysis of co-occurrence data. It can be adopted to construct low dimensional latent space from observed document-word pairs. Let the pair (w, d) denotes the co-occurrence of a word, w, and a document d, PLSI models the joint probability as $\mathbb{P}(w, d) = \mathbb{P}(d) \sum \mathbb{P}(z|d)\mathbb{P}(w|z)$, where z is a latent topic variable. Since PLSI has been extensively used in data mining for topic clustering and representation of latent space, we analyze commodity description using PLSI model. We consider each item description as a document which consists of a number of words. After PLSI modeling, both item and word can be represented using a latent vector $z = \{z_i | i \in \mathbb{I}\}$ under the constraint that $0 \leq z_i \leq |\mathbb{I}|$ and $\sum z_i = 1$, where $\mathbb{I}$ is the set of latent topics.

We use the vectors in the topic space to calculate the product similarity. Similarity is derived from Euclidian distance. The vectors are first normalized to length 1 and Euclidian distances between any two points are derived. Given distance d, the similarity is formulated as $\frac{1}{1+d}$. Conventional information retrieval methods often use cosine similarity (or cosine angle distance) to represent how similar two points are [reference]. Qian et al. found that cosine angle distance and Euclidean distance perform similarly while dimension is high (larger than 128). Moreover, Euclidean distance exhibits triangle inequality; thus ease the design of pruning strategy of calculating inter-similarity between products.

Pruning Strategy. The time and space complexity of calculating the inter-similarity among N product is $O(N^2)$. Since our E-Commerce site has more than 50,0000 products. Calculating inter similarity could be a tedious work without pruning strategy. To make this problem tractable, we use k-means to divide those items into m clusters, where $m \ll N$. Given any product p and its belonging cluster, we can obtain the distance between p and center of the cluster while conducting k-means clustering. Furthermore, we calculate the pairwise-similarity of those m cluster centers. Given those distances and a user specified minimum threshold of similarity. We can leverage triangle inequality to decrease the computational cost. In a metric space, given any three points x, y, z, we have triangle inequality:

$$d(x,z) \leq d(x,y) + d(y,z) \tag{1}$$

where $d(x,y)$ denotes the distance between two points x and y. We can derive the following relation from (1):

$$d(x,y) \geq |d(x,z) - d(y,z)| \tag{2}$$

Given any two products p_1 and p_2 belongs to cluster c_1 and c_2 respectively. The distances from c_1 to c_2, p_1 to c_1, and p_1 to c_2 are $d(c_1,c_2)$, $d(p_1,c_1)$, and $d(p_2,c_2)$, respectively. Based on triangle inequality, $d(p_1,p_2) > |d(p_1,c_2) - d(p_2,c_2)|, d(p_1,c_2) < |d(c_1,c_2) - d(c_1,p_1)|$; therefore

$$d(p_1,p_2) > ||d(c_1,c_2) - d(c_1,p_1)| - d(p_2,c_2)| \tag{3}$$

In our system, we want to find product pairs with high similarities. So it is reasonable use a user specified minimum similarity threshold s. Given s, the maximum distance d allowed between two points can be derived. In (3), we can obtain the lower bound of distance dl from p_1 to p_2. If $dl > d$, that means the similarity between p_1 and p_2 definitely smaller than s, we can safely ignore this pair. We built our pruing strategy by utilizing a common open source machine learning library: Mahout [7], while using its built-in functions to calculate iter-item to calculate similarities.

2.4 Filtering Candidates with High Similarity

In our proposed method, we use algorithms similar to find VV and BB pairs to find the candidates of complementary product catalog tree nodes. For each pair of tree nodes N_A and N_B, we derive $\mathbb{P}(N_A), \mathbb{P}(N_B)$ and $\mathbb{P}(N_A, N_B)$, and use F to obtain the value W represents the correlation between N_A and N_B. It is just like what we have done while finding VV or BB pairs. We use a threshold T for W to filter the nodes with high correlation as candidate pairs. Given each candidate nodes pair N_A and N_B, if products contained in N_A and N_B are very similar, N_A and N_B are not complementary. To eliminate those nodes, for all pairs of products where one product belongs to N_A and the other belongs to N_B, we calculate the average similarity. Nodes are regarded as complementary if the average similarity smaller than a specified threshold s.

3 Experiments

In this section, we present experiment result demonstrating the performance of our algorithms.

3.1 Filter Based Algorithm

Our first filter based algorithm requires editors to hand pick complementary nodes. From root to leaves, there are 5 levels in our product catalog tree. We list the manually specified nodes complementary to node clothes and underwear. To avoid tedious manual work, the complementary nodes are all nodes in 1st level (1 degree from root) of product catalog tree. In our experiment, we choose clothes as our major node. Editors identified the following categories which are complements to the clothes node:

1. Watch, Jewellery & Accessories I
2. Watch, Jewellery & Accessories II
3. Shoes & Bags
4. Accessories

The item coverage is about 11.3 %, meaning that we derive complementary products for 11.3 % products on our e-commerce site. We also integrate this model into an online recommender module, which recommend complementary and bought together items in the shopping cart as shown in Fig. 1. The previous two recommended items (shoes) are complementary with the item (dress) in the shopping cart. The average CTR of this module is 7.7 % and the average GMS of the module is 0.46 %.

A sample of our experiment results is listed in Table 1. The first column denotes the node that are complementary to clothes. In the column denoted by “Complementary Items Examples”, the first item is the sample clothing and the second items is the complementary item for the sample clothing.

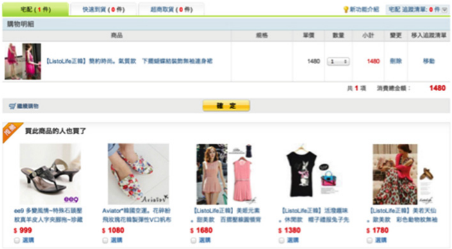

宅配 (1 件)
快速到貨 (0 件)
超商取貨 (0 件)
新功能介紹
宅配 追蹤清單: 0 件
購物明細
商品
規格
單價
數量
小計
變更
移入追蹤清單
【ListoLife正韓】簡約時尚。氣質款 下擺蝴蝶結裝飾無袖連身裙
1480
1
1480
刪除
移動
共 1 項
消費總金額：
1480
繼續購物
確 定
推薦
買此商品的人也買了
ee9 多變風情~特殊石頭壓紋真羊皮人字夾腳拖~珍藏
$ 999
選購
Aviator*韓國空運。花碎粉飛玫瑰花韓製彈性V口帆布
$ 1080
選購
【ListoLife正韓】美姬元素。甜美款 百摺壓線圓領背
$ 1680
選購
【ListoLife正韓】活潑趣味。休閒款 帽子纏服兔子先
$ 1380
選購
【ListoLife正韓】美若天仙。歐美款 彩色動物紋無袖
$ 1780
選購

Fig. 1. Complementary recommendation on a typical page

3.2 Find Complementary Nodes Automatically

Since the second method can find complementary nodes without manual work, we use nodes in 2nd level (2 degree from root) in the product catalog tree. Since the products in 2nd level nodes are more homogeneous than the 1st level, we can explore complementary products more precisely. We first use the buying event during May, June, and July 2012, to find the candidates of complementary nodes. Afterwards, we eliminate pairs of nodes with high similarities. The table below illustrates the candidate complementary nodes of the node "COACH", which containing the bags and accessories of brand COACH. We use 0.8 as a threshold and eliminate the nodes with similarities greater than 0.8 will be eliminated. By doing so, we eliminated the nodes containing luxury brands bags.

We also have the editor to verify the precision of complementary nodes predicted by our topic mining model. We list the verification results in Table 3. The first column lists the product clusters we used as the seed for prediction. The second column is the precision, which is the proportion of nodes regarded as complementary by the editor over the total numbers of nodes that are generated by our algorithm. The proportion of predicted complementary nodes that are actually deemed by the editors as similar nodes are listed in the third column. The proportion of predicted complementary nodes that are actually irrelevant is listed in column four.

4 Discussion

In our proposed method, we first specify the complementary nodes manually, and find complementary items consistent in style by using collaborative filtering base method. This strategy can achieve high precision but requires tedious manual work. Therefore it is not scalable. The second proposed method is to derive the complementary nodes from products bought together. Considering besides complementary products, there're some similar products bought together, we have to eliminate nodes containing similar products. Therefore, a pruning strategy is developed for calculating inter-similarity among large amount of products. And successfully eliminates similar nodes. In our experiment, we found except similar and complementary products, there are also some products seems irrelevant are

Table 1. Example of items with complementary relation

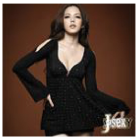
Node Name	Complementary Items Examples	
Watch, Jewery & Accessories I		
	casual style dress	silver necklace
Watch, Jewery & Accessories II		
	casual style dress	silver necklace
Shoes & Bags		
	dress	high heels
Accessories		
	dress	high heels

Table 2. complementary product clusters with respect to the cluster Coach

Node name	Similarity	Eliminated
Trendy accessory	0.75	
European luxury brands bags	0.81	Y
Casual style shoes	Below 0.75	
Perfume	Below 0.75	
Clothes and underwear	Below 0.75	
GUCCI	0.81	Y
LV	0.80	Y
USA and Japan luxury brands bags	0.80	Y
Luxury brands accessories	0.76	
Watches	Below 0.75	

Table 3. Precision of complementary node prediction from our topic mining model with error types.

Node name	Complement nodes	Similar nodes	irrelevant nodes
Computer	56 %	5 %	39 %
Trendy lady's clothes	43 %	16 %	31 %
Trendy lady's shoes	68 %	6 %	26 %
European luxury brands bags and accessories	50 %	12 %	38 %

bought together, which makes some derived complementary nodes not precise. Currently we have not find a good strategy to eliminate those node pairs. The editors have to remove those nodes manually (Table 2).

5 Future Work

The future work of complementary product finding could be can be divided into 2 aspect. The first one is extending the characteristic of complementary products such as the similarity between two complementary products, the penalty obtain from popularity index to finding complementary media contents. The second aspect is we can leverage complementary products to facilitate the recommender system in small business, product search or Ad targeting.

References

1. Strands Labs, I.: Best practices for product recommendations on ecommerce websites. Technical report, Strands Labs, Inc (2010)
2. Amit, R., Zott, C.: Value drivers of e-commerce business models. Number 2000–2006. INSEAD (2000)
3. Zott, C., Amit, R., Donlevy, J.: Strategies for value creation in e-commerce: best practice in Europe. Eur. Manag. J. **18**(5), 463–475 (2000)
4. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: ACM SIGMOD Record, vol. 22, pp. 207–216. ACM (1993)
5. Linden, G.D., Jacobi, J., Benson, E.: Collaborative recommendations using item-to-item similarity mappings. United States Patent (1998)
6. Bowman, M., Debray, S.K., Peterson, L.L.: Reasoning about naming systems. ACM Trans. Program. Lang. Syst. (TOPLAS) **15**(5), 795–825 (1993)
7. Apache Foundation: The apache mahout machine learning library. http://mahout.apache.org/

A Short-Term Bookmarking System for Collecting User-Interest Data

Chu-Cheng Hsieh(✉), Yoni Medoff, and Naren Chittar

EBay, Inc., 2065 Hamilton Ave, San Jose, CA 95125, USA
{chsieh,ymedoff,nchittar}@ebay.com

Abstract. During the shopping process, users typically narrow down their search to a small collection of products before making a final purchase. These data, consisting of products that users are considering purchasing, correlate strongly with user search intent and product desirability. By allowing users to bookmark products between browsing and purchasing, we collect user-interest information. We then propose a product recommendation algorithm based on these data. By considering both popular and long-tail queries, we shed light on the potential usage of the data.

Keywords: User interface · Recommendation system · Collaborative filtering

1 Introduction

Product recommendation have been a popular feature of e-commerce sites for helping buyers find what they want. A common practice is to recommend products based on user behavior, such as "people viewing/buying this are also viewing/buying that." Developing such a recommendation system is very challenging for a site such as eBay. Applying "buy-together" is not feasible because many products are unique with respect to either price or condition (or both). On the other hand, "view-together" often requires further offline processes, for instance, to break down user sessions into meaningful segments of intent, or to parse logs for finding similar products (belonging to the same cluster [6]). Furthermore, data collected from user activity logs often contain a certain degree of noise.

One simple solution is to collect information directly from the users, that is, asking each user about his shopping intent and the products he is considering. However, mandating users to provide feedback is often very inconvenient for the user, and may cause negative impact in user shopping experience. In this work, we propose a roundabout — we collect "user-interest data" by providing a short-term bookmarking mechanism. Our approach gives buyers a better shopping experience by helping them minimize the inconvenience of switching between search result pages and product viewing pages. At the same time, the proposed prototype helps us in collecting the aforementioned data.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 247–254, 2014.
DOI: 10.1007/978-3-319-13186-3_24

Note that data collected by our proposed system are very different from data acquired from wishlists or collections.[1] Of course, products that are collected together usually share some similar characteristics, for example consider the collections entitled *baby stuff* and *new home.* Often the intent of creating these lists are quite broad. For instance, a cradle could be assigned to *baby stuff* or *new home.* Although further processes such as topic modeling techniques [3] could be applied in providing better recommendations by clustering, we would still face the well-known "cold start" problem, that is, to infer a right decision, one needs to wait until sufficient data has been gathered and analyzed. Since creating a wishlist/collection is an optional behavior in the shopping process at an e-commerce site, many popular or desirable deals might be gone by the time we collect enough data and complete our offline analysis.

Under the hood, our proposed solution is based on crowd wisdom and frequent itemset mining. We believe that, given search result pages of a query, products being considered by many users are often active listings and "good deals." That is to say, overpriced, outdated, or suspicious listings are naturally filtered out by users. Next, through applying association rule learning, we can easily make timely recommendations based on these high quality data. Our work collects data that fills the gap between view-action and buy-action. It empowers c2c sites to provide in-session personalization — the user's recommendations are affected by the products bookmarked in real-time.

2 Collecting User-Interest Data

During the course of shopping, users often collect a small assortment of products that they are considering purchasing. They deliberate over this pool of products, and often choose to purchase one from among them. There are many ways that users can accomplish this behavior on current e-commerce sites, such as opening multiple tabs, saving links, or going back and forth between product pages and search result pages. However, these methods can be tedious for many users, and the logs from such activities are not always directly related to user-interest levels. Clicks and impressions are sometimes used to infer user-interest data, but it would be much more accurate to collect interest data explicitly from the user. There is the additional challenge that any such system must also present significant value to the user, so that he has some motivation to actively use the system when shopping on an e-commerce site.

We have developed a short-term bookmarking system which serves as one possible method to facilitate the collection of user-interest data. This acts as a method for users to actively collect a small number of products on eBay. The core part of the interface is comprised of an interactive portion at the bottom of the screen. Whenever a user is interested in a certain product on the site, he simply drags the image of that product into this interface, as shown in Fig. 1.

[1] Collections enable users to bookmark products and organized in one place, for example, http://www.ebay.com/cln, or http://www.pinterest.com.

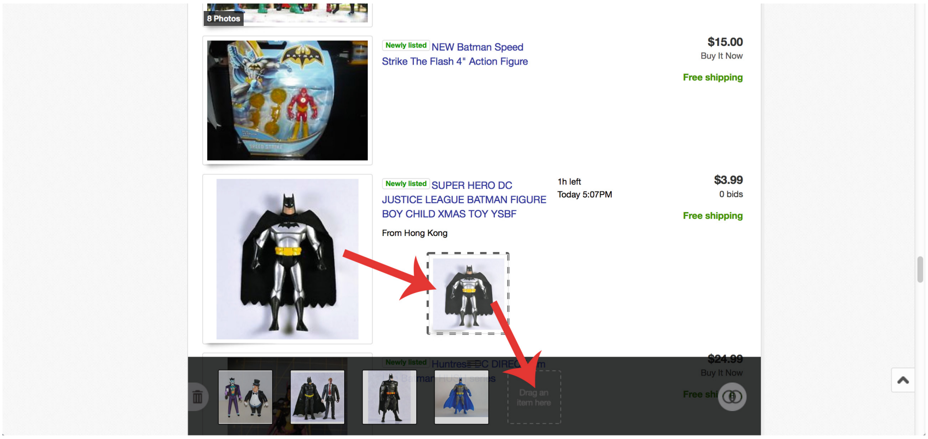

8 Photos
Newly listed NEW Batman Speed Strike The Flash 4" Action Figure
$15.00
Buy It Now
Free shipping
Newly listed SUPER HERO DC JUSTICE LEAGUE BATMAN FIGURE BOY CHILD XMAS TOY YSBF
From Hong Kong
1h left
Today 5:07PM
$3.99
0 bids
Free shipping
Newly listed Huntress DC DIREC
$24.99
Buy It Now
Drag an item here

Fig. 1. Drag-and-drop bookmarking

Our interface allows him to easily bookmark such products, which are always available for navigation from the strip of thumbnail images.

This system also features an intuitive user interface to ensure significant usefulness while maintaining a minimal footprint. The container that houses these products is fairly intelligent in responding to user behavior on the page. When the user scrolls down, the container minimizes to get out of the user's viewing area (toward the bottom of the screen). If the user scrolls up, or attempts to interact with the interface by dragging a product image, the container will shift back into view. In this way, the system presents itself only when necessary.

Our system limits the maximum number of products a user can bookmark. This limitation forces the user to keep a small pool of products for which he has expressed interest, and it maintains data quality by ensuring that each product selection is important in making purchasing decisions for a single session. This system is also designed to be persistent across browsing sessions and devices. All bookmarked products remain unless explicitly deleted by the user. This functionality has two main advantages: (1) the user can work from multiple devices or browsers while retaining the same bookmarks, and (2) we can observe deletion behavior, which can coincide with intent, such as "making room" for better product choices.

Basic interaction with our system allows us to accurately predict user-interest data in real-time. The system allows for many combinations of user actions, which we can interpret as varying levels of interest, especially in accordance with other elements of the page and other actions on the site. For example, if a user adds an product and then quickly removes it, we can infer a low or superficial level of interest for that product. If a user removes the last product A, and then quickly replaces it with another product B, we can infer that the interest level in B supersedes that of A. By aggregating this type of data over many users, and combining it with purchasing behavior, search queries, browsing patterns, etc., we can form very robust algorithms for product recommendations, and even search relevance in general.

3 Product Recommendations

In this section, we discuss our product recommendation algorithm. The algorithm consists of the following steps:

1. Convert the bookmarks into transactions
2. Cluster transactions by query intent
3. Use association rule mining techniques to derive recommendation candidates
4. Rank candidates based on support, confidence, and other measurements

3.1 Algorithm

We now discuss our recommendation algorithm. Theoretically, any association rule mining techniques [2,12] could satisfy our need to generate recommendations. However, due to performance concerns for e-commerce sites like eBay, we always choose algorithms [7] that could be implemented using the map-reduce programming model.

Let $t_x = \{i_1, i_2, ..., i_m\}$ be an itemset where each item corresponds to one product (bookmarked by drag-and-drop), and let D_k represent the collection of all bookmarks that share the same intent k. One simple method is to consider every t as a transaction and every query as an intent cluster D_k. For every cluster, we run an association rule mining algorithm, and all rules are associated with corresponding *Support* and *Confidence* [1] measurements.

Assuming a user bookmarks three items $\{A, B, C\}$, we seek association rules for which all items exist in the *antecedent* (left-hand-side), for example, $\{A, B, C\} \Rightarrow \{D\}$. Then any item(s) in the *consequent* (right-hand-side) are considered legitimate candidates. Intuitively, when the antecedent of an association rule matches exactly the items in a bookmark, every item in the consequent becomes a member of the candidate set. We rank those candidates by

$$- log\frac{\mathcal{C}}{\mathcal{C}+\mathcal{S}} - log\frac{\alpha * \mathcal{S}}{\mathcal{C}+\mathcal{S}} \tag{1}$$

This formula refers to the Shannon entropy [10] except that we introduce α as a tuning parameter of popularity (support, denoted by $\mathcal{S}$) and accuracy (confidence, denoted by $\mathcal{C}$).

We have adopted two main strategies to control the size of the candidate set. First, by adjusting the threshold of support and confidence, we can increase or decrease the size of the candidate set. Sometimes, especially when a user bookmarks many products, it may be problematic to continue lowering the thresholds of support and confidence in order to find candidates. In such a scenario, we conduct soft-matching — the antecedent matches only a subset of bookmarked products. Namely, if we have 7 products bookmarked, we seek association rules that contain at least, say, 6 products in their antecedents. The score of a candidate (for ranking) becomes

$$- n * [log \frac{\mathcal{C}}{\mathcal{C} + \mathcal{S}} + log \frac{\alpha * \mathcal{S}}{\mathcal{C} + \mathcal{S}}] \tag{2}$$

where n represents the number of matched rules for the same candidate. For example, if we have two association rules $\{A, B\} \Rightarrow \{E\}$ and $\{B, C\} \Rightarrow \{E\}$, and a user bookmarks three items $\{A, B, C\}$, the score of the candidate item E would be doubled ($n = 2$) in a soft-matching case. Note that in soft-matching cases, if there are better matches, for instance the rule $\{A, B, C\} \Rightarrow \{D\}$, we assign n to the number of possible combinations, i.e. it matches $\{A, B\}$, $\{A, C\}$, and $\{B, C\}$.

3.2 Extension

In this section, we discuss two extensions for increasing recommendation quality — one targets popular queries, and the other targets long-tail queries.

In a short time windows, for a given query q, products returned by a search ranking algorithms are very similar, partly because there is often a delay to pass inventory data in transactional databases into inverted indices (a central component of a typical search engine indexing algorithm). For popular queries like "iPhone," in a short time window, say 30 min, the system probably collects hundreds or thousands of bookmarks to make recommendations. Therefore, constraining association rule minings to a time window for popular queries ensures the products seen by users are mostly identical. It may also boost the chance of bookmarking the same candidates, because according to the study conducted by Granka et al. [5], often products with the highest rankings draw the most attention. Therefore, products in the first page become an indicator to learn user preferences.

For e-commerce sites, long-tail queries are often difficult to handle because they require a long period of time in order to acquire enough data for complete analysis. So far, we assume that a user intent (and its corresponding cluster D_k) is constrained or represented by a user query, and this assumption becomes problematic for long-tail queries.

We address long-tail queries by studying query transition (i.e. how users reform their queries). Query transition has been studied extensively in the past and has shown its success in helping query reformulation [4]. To address long-tail queries, we rely on their preceding queries. For example, we enlarge the cluster of the query "iPhone 5 Gold 64 GB" by considering both "iPhone 5 Gold" and "iPhone 5 64 GB."

We apply the notion of probability matching here for long-tail queries. Assuming a query transition graph is provided where every vertex corresponds to a query, and a directed link from query a to q is associated with a probability P_{aq}, referring to the probability that a user reforms a into q. We identify a set of preceding queries $\mathcal{A}$ where $P_{aq} \geq \theta_p$, $a \in \mathcal{A}$, and the query a contains an adequate number of reliable recommendations, i.e. without applying soft-matching.

θ_p is usually a parameter of controlling diversity — if θ_p is smaller, the number of preceding queries $\mathcal{A}$ is larger. We then randomly select a preceding query x in proportion to its probability P_{xq}, i.e. $P_{xq}/\sum_{a\in\mathcal{A}} P_{aq}$. The first time a preceding query x is selected, we draw the first recommendation based on Eq. 2, and the next time we draw the second one, and so on. This probability matching ensures diversity in guessing user intention based on transition probability. A screen shot of our recommendations is shown in Fig. 2.

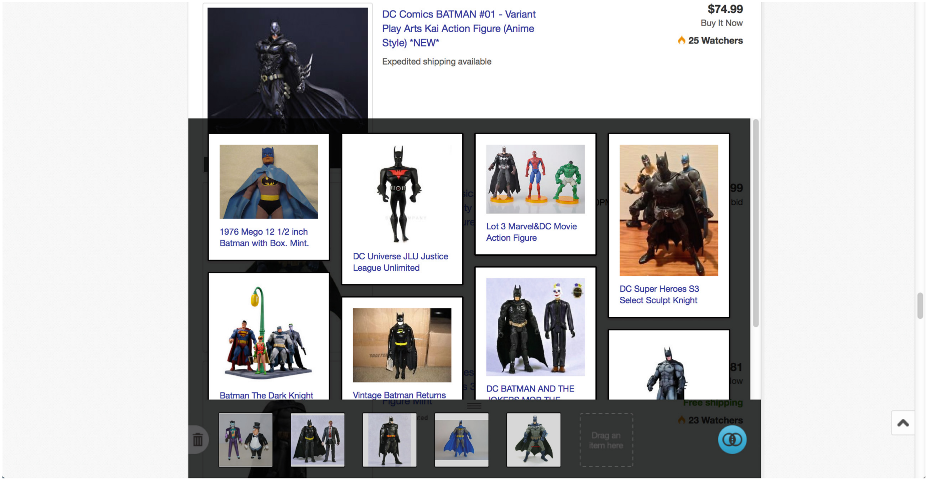

DC Comics BATMAN #01 - Variant Play Arts Kai Action Figure (Anime Style) *NEW*
Expedited shipping available
$74.99
Buy It Now
25 Watchers
1976 Mego 12 1/2 inch Batman with Box. Mint.
DC Universe JLU Justice League Unlimited
Lot 3 Marvel&DC Movie Action Figure
DC Super Heroes S3 Select Sculpt Knight
Batman The Dark Knight
Vintage Batman Returns
DC BATMAN AND THE
Free shipping
23 Watchers
Drag an item here

Fig. 2. Displaying recommendations to the user

4 Related Work

Product recommendation has been studied extensively in the past. Linden et al. [8] focused on long-living products based on co-purchasing behavior. Katukuri et al. [6] used clustering algorithms in recommending similar products. Later Xiao et al. [11] elicited the interests of individual customers, and Park et al. [9] used individual/group profiling to generate personal recommendations. Most of the work related to production recommendation or user preferences are based on analyzing data in logs or inventory. Our work differs in the methodology for both collection and analysis. We gather user-interest data through explicit user actions. While other works infer user-interest from related data, we have devised a system which naturally encourages the user to direclty produce these data. Secondly, we emphasize the collection of these data in real-time and the potential applications that arise from real-time feedback.

5 Conclusion

In this work, we design an interactive bookmarking system that both improves the e-commerce shopping experience and provides valuable user-interest data for researchers. This system responds to a users natural inclination during shopping to collect a small number of products in making a purchasing decision. It facilitates the gathering of valuable data on user-interest levels in particular products relative to other aspects of shopping behavior — which product users are considering. These data, aggregated over many users, can be applied to many important problems. Moreover, compared to logs that require post-processing, the user-interest data are cleaner and can therefore be processed in real-time.

Acknowledgement. We appreciate the help from Noah Batterson for his work in designing the user interface of this project. In addition, we thank Lan Wang for sharing her insights and help in building the first prototype. Our work is greatly influenced by their knowledge of improving user experience.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: 19 ACM SIGMOD Conf. on the Management of Data. Washington, DC (May 1993)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In International Conference On Very Large Data Bases (VLDB '94), pp. 487–499. Morgan Kaufmann Publishers Inc., USA (Sept. 1994)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
4. Boldi, P., Bonchi, F., Castillo, C., Donato, D., Vigna, S.: Query suggestions using query-flow graphs. In: Proceedings of the 2009 Workshop on Web Search Click Data, WSCD '09, pp. 56–63. ACM USA (2009)
5. Granka, L.A., Joachims, T., Gay, G.: Eye-tracking analysis of user behavior in www search. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '04, pp. 478–479. ACM, USA (2004)
6. Katukuri, J., Mukherjee, R., Konik, T.: Large-scale recommendations in a dynamic marketplace. In: Workshop on Large Scale Recommendation Systems at RecSys'13 (2013)
7. Lin, M.-Y., Lee, P.-Y., Hsueh, S.-C.: Apriori-based frequent itemset mining algorithms on mapreduce. In: Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, ICUIMC '12, pp. 76:1–76:8. ACM, USA (2012)
8. Linden, G., Smith, B., York, J.: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2003)
9. Park, Y.-J., Chang, K.-N.: Individual and group behavior-based customer profile model for personalized product recommendation. Expert Systems with Applications, 36(2, Part 1):1932–1939 (2009)
10. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J., 27:379–423, 623–656 (1948)

11. Xiao, B., Benbasat, I.: E-commerce product recommendation agents: Use, characteristics, and impact. MIS Q. **31**(1), 137–209 (2007)
12. Zaki: Scalable algorithms for association mining. IEEETKDE: IEEE Trans. Knowl. Data Eng. 12 (2000)

Cloud Service Discovery

Automatic Self-Suspended Task for a MapReduce System on Cloud Computing

Tzu-Chi Huang[1](✉), Ce-Kuen Shieh[2], Sheng-Wei Huang[2], Chui-Ming Chiu[2], and Tyng-Yeu Liang[3]

[1] Department of Electronic Engineering,
Lunghwa University of Science and Technology,
Guishan, Taoyuan County, Taiwan
tzuchi@mail.lhu.edu.tw

[2] Institute of Computer and Communication Engineering,
Department of Electrical Engineering,
National Cheng Kung University, Tainan, Taiwan
{shieh, swh, cmchiu}@ee.ncku.edu.tw

[3] Department of Electrical Engineering, National Kaohsiung University of Applied Science, Kaohsiung, Taiwan
lty@mail.ee.kuas.edu.tw

Abstract. A MapReduce system gradually becomes an essential technology to achieve the large scale computing on cloud computing. A MapReduce system currently is designed to distribute tasks over nodes in a cloud according to manual configurations of slot numbers in nodes. However, a MapReduce system may have the performance degradation due to the inappropriate configuration of the slot number, because the slot number can not exactly reflect the performance of the node. A MapReduce system can utilize the Automatic Self-Suspended Task (ASST) proposed in this paper to alleviate the performance degradation due to the inappropriate configuration of the slot number in a node on cloud computing. In experiments of this paper, a MapReduce system is proved to have a better performance with the help of ASST for various applications on cloud computing.

Keywords: MapReduce · Cloud computing · ASST · Automatic Self-Suspended Task

1 Introduction

A MapReduce system [1] gradually becomes an essential technology to achieve the large scale computing on cloud computing [2, 3]. A MapReduce system attracts programmers by automatically handling issues about distributed and parallel programming on behalf of the programmers, so they can focus on the development of applications. A MapReduce system can work for applications easily as only as programmers develop their applications with a Map function and a Reduce function. At run time, a Map Reduce system automatically distributes tasks of an application over nodes in a cloud across networks and relies on the tasks to execute Map and Reduce functions. Finally, a MapReduce system collects partial results from nodes in a cloud and merges the results as the application output.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 257–268, 2014.
DOI: 10.1007/978-3-319-13186-3_25

Proposed by Google and according to its open-source implementations such as Hadoop [4], a MapReduce system currently defines the node computation capability as manually configured slot numbers and distributes tasks over nodes according to the slot numbers. When a task is required to create for executing a Map or Reduce function (i.e. a Map or Reduce task), a MapReduce system generally finds the node that has the maximum free slot number, besides considering data locality of input data intra or inter racks for Map tasks. A MapReduce system assumes that the node having the maximum free slot number should be the idlest node among all nodes in a cloud. However, a MapReduce cannot work well according to the manual configuration of the slot number.

Basically, a slot number cannot be configured easily as a static value to get the best performance for a MapReduce system, e.g., when a cloud simultaneously runs tasks of multiple MapReduce applications or early pipelines [5] the movement of intermediate data from Map tasks to Reduce tasks instead of doing the movement in a specific shuffle phase. Moreover, a slot number cannot directly correspond to the node computation capability, especially in a heterogeneous environment composed of multiple nodes that have different hardware components. A slot number usually is configured according to the number of CPU cores in a node. However, a slot number in a node having a low speed CPU is not comparable to that in a node having a high speed CPU. Even though all nodes in a cloud have the same CPU, a slot number still cannot be directly used to evaluate the node computation capability because other hardware components such as memory size, memory speed, and hard disk speed, may affect the performance as well. Furthermore, a slot number cannot reflect the performance of a node at run time because different tasks may compete with each other for resources to affect the performance. A slot number may be inappropriately configured to have a seriously negative impact on the performance of a MapReduce system, but not many related works propose solutions against the performance degradation stemming from the manual configuration of the slot number in a node on cloud computing.

In this paper, the Automatic Self-Suspended Task (ASST) for a MapReduce system is proposed to alleviate the performance degradation stemming from the inappropriate configuration of the slot number in a node on cloud computing. ASST can work for a MapReduce system and maintains a good performance regardless of the configuration of the slot number in a node on cloud computing. ASST turns each of Map and Reduce tasks into a task capable of suspending the execution on demand without degrading the performance when many computation-intensive tasks compete with each other for CPU time. ASST uses the prediction model of CPU utilization trend to determine whether running or suspending a task is good to performances at run time. ASST can be applied to a cloud where Map and Reduce tasks are executed simultaneously to get the pipeline MapReduce benefits [5] or tasks of multiple applications are executed simultaneously to improve resource utilization. While today most MapReduce applications are computation-intensive, ASST in this paper brings contributions to cloud computing as follows.

- This paper is the first paper proposing a task self-suspension mechanism (i.e. ASST) to improve the performance of a MapReduce system on cloud computing.
- This paper introduces a prediction algorithm in ASST to predict the CPU utilization in a node in order to determine whether the task execution should be continued or not.

- This paper proposes the prediction model of CPU utilization trend to determine whether running or suspending a task is good to performances at run time.
- This paper implements the proposal of ASST in a MapReduce system in Windows 2003 and makes it compatible to a task scheduler used by Hadoop [4].
- This paper conducts experiments with several popular applications on cloud computing to verify its practicability.
- This paper proves that ASST indeed can improve performances of applications in comparison to the applications that do not use ASST.

This paper is organized as follows. Section 2 reviews MapReduce. Section 3 presents Automatic Self-Suspended Task (ASST). Section 4 presets the ASST implementation and experiment results. Section 5 concludes this paper.

2 MapReduce

MapReduce is a programming model proposed by Google in order to process large datasets in a cloud. MapReduce is composed of a Map function and a Reduce function. MapReduce uses a Map function to process input data before a Reduce function can be executed to process intermediate data produced by a Map function. MapReduce provides a Map function with input data but a Reduce function with intermediate data composed a series of key and value pairs. Finally, MapReduce expects a Reduce function to merge values associated with the same key. MapReduce is often explained with Word Count [1], a typical application on cloud computing.

Word Count uses a Map function to scan input data for each word separated by space characters. In a Map function, Word Count emits a key/vale pair as a record of intermediate data. Take a word “apple” in input data, for example, Word Count emits a string “apple” as the key and a value “1” as the value in intermediate data cached in an intermediate file. During the execution of Map functions, Word Count relies on a MapReduce system to both shuffle intermediate data among nodes and create tasks to execute Reduce functions in nodes for processing the corresponding intermediate data. In a Reduce function, Word Count merges values associated with the same key. Take intermediate data having a series of “apple” and “1” as its keys and values for example, Word Count sums up the numbers of value “1” as the count of word “apple” in input data.

At run time, a MapReduce system finds the suitable node responsible for running a Map task (i.e. a task executing a Map function) and provides the Map task with input data loaded from a disk in a Master node or a distributed file system such as Hadoop Distributed File System (HDFS) [6]. While finding a node to run a Map task, a MapReduce system may choose the node close to the location of input data, e.g. nodes in a rack having input data to avoid moving input data from a node to another node across racks. While intermediate data is produced by Map tasks, a MapReduce system finds a suitable node responsible for running a Reduce task and moves intermediate data to the suitable node. After all Reduce tasks finish execution, a MapReduce collects and merges their results as the application output.

A MapReduce system such as Hadoop (an open-source implementation of Google MapReduce) chooses a suitable node according to basic principles as follows.

- Finding a node that has the maximum free slot number and transferring input data from nodes in the same rack to the node for running a Map task.
- Finding a node that has the maximum free slot number in other racks if a suitable node coexisting with nodes having input data in the same rack can not be found.
- Finding a node that has intermediate data and a free slot number to run a Reduce task in order to avoid consuming network bandwidth of intermediate data movement.

Accordingly, a MapReduce system chooses a node for running a Map task or a Reduce task strongly according to the configuration of the slot number in a node. If a node is configured to have more slot numbers, a MapReduce system probably assigns more tasks to the node and has a rick of overloading it to degrade the overall performance. A MapReduce system can use ASST proposed in this paper to alleviate the negative impact on the performance stemming from the inappropriate configuration of the slot number in a node on cloud computing.

3 Automatic Self-Suspended Task (ASST)

3.1 Overview

Automatic Self-Suspended Task (ASST) is an enhanced Map or Reduce task capable of being temporarily suspended by itself at run time in order to give other tasks more CPU resources to finish their works. ASST separates tasks of a MapReduce application into Map tasks and Reduce tasks, and works inside each of the tasks respectively. ASST may determine that a Map task should be suspended but keeps a Reduce task running. Because ASST resides in each task and just works for the task as shown in Fig. 1, so not all tasks belonging to the same type will be run or suspended at the same time. ASST uses a prediction model of CPU utilization trend in order to determine to keep a task in the current status or not, i.e. switching to another status or keeping the status intact. ASST defines a task to have either of the running status or the suspended status (temporarily). According to the prediction model of CPU utilization trend, ASST switches the status of a task between the running status and the suspended status at checkpoints inserted inside Map functions and Reduce functions to not only efficiently utilize resources but also avoid drawbacks due to resource competition among tasks.

For example, a MapReduce system may keep assigning Map tasks to a node until reaching the limitation of the Map task slot number if Map tasks finish their works quickly. When Map tasks produce intermediate data, a MapReduce system consequently has much intermediate data waiting for being processed and creates more Reduce tasks at nodes to consume intermediate data. Accordingly, a MapReduce system may overload a node by many tasks, especially due to the inappropriate configuration of the slot number in a node. Conversely, a MapReduce system may create Reduce tasks much more than Map tasks and cannot fully utilize Reduce tasks, so most Reduce tasks may be created just for processing very few intermediate data and

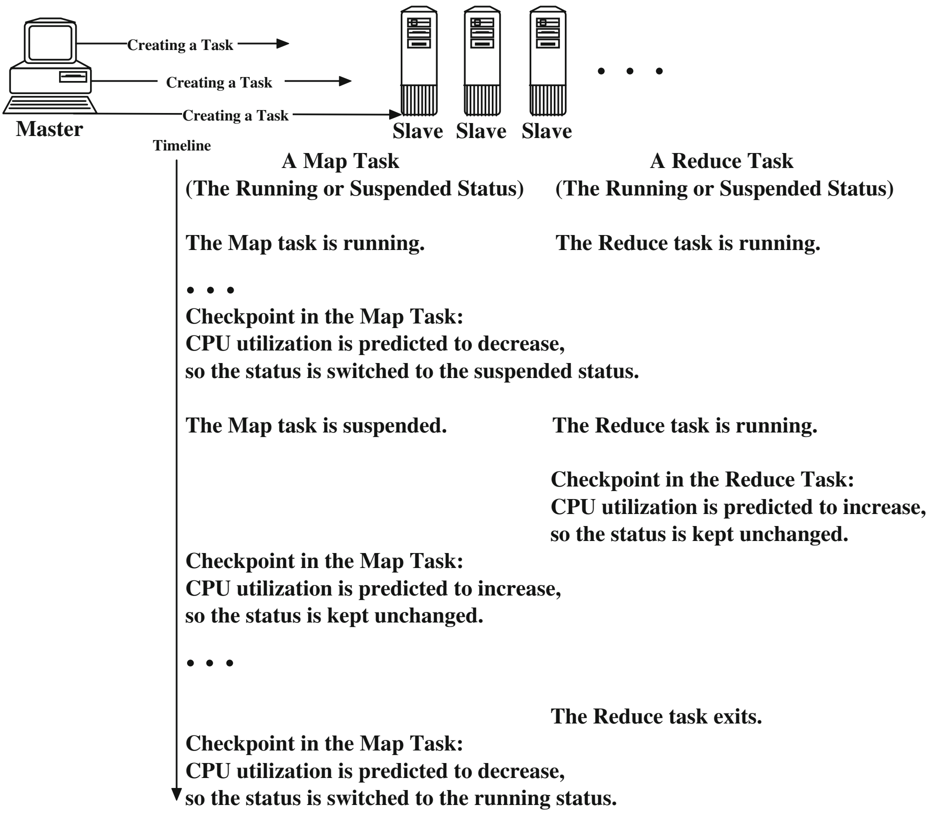

Master
Creating a Task
Creating a Task
Creating a Task
Slave
Slave
Slave
• • •
Timeline
A Map Task
(The Running or Suspended Status)
A Reduce Task
(The Running or Suspended Status)
The Map task is running.
The Reduce task is running.
• • •
Checkpoint in the Map Task:
CPU utilization is predicted to decrease,
so the status is switched to the suspended status.
The Map task is suspended.
The Reduce task is running.
Checkpoint in the Reduce Task:
CPU utilization is predicted to increase,
so the status is kept unchanged.
Checkpoint in the Map Task:
CPU utilization is predicted to increase,
so the status is kept unchanged.
• • •
The Reduce task exits.
Checkpoint in the Map Task:
CPU utilization is predicted to decrease,
so the status is switched to the running status.

Fig. 1. ASST overview

then exit. According to the inappropriate configuration of the slot number in a node, furthermore, a MapReduce system may make the application work always in a workload unbalance condition by either producing intermediate data too fast or consuming it too slow. In brief, a MapReduce system can work with ASST to avoid drawbacks of the inappropriate configuration of the slot number in a node on cloud computing.

3.2 Working Principle

According to [7], CPU utilization will increase in proportional to the degree of process number in an operating system. However, CPU utilization will decrease suddenly when the process number reaches a certain degree. At that time, CPU utilization can come back to the better level only if the degree of process number is decreased. Since Map and Reduce tasks are created at a node by a Master node to work with each other in a manner similar to the relation between producers and consumers, CPU utilization may be kept high with a certain level of competition fallback between Map tasks and Reduce tasks.

We propose to use ASST for determining whether Map or Reduce tasks should be kept running or suspended according to the prediction model of CPU utilization trend. If keeping Map tasks in the running status can get the positive result from the prediction model of CPU utilization trend, we run Map tasks continuously. If keeping Map tasks in the suspended status can get the positive result from the prediction model of CPU utilization trend, we suspend Map tasks gradually. Similarly, we work on Reduce tasks by gradually keeping them in the running status or the suspended status according to the results from the prediction model of CPU utilization trend. We do not limit a Map task or a Reduce task to which status, and we only change the status of a task once predicting the trend of CPU utilization toward degradation.

Equation to Predict CPU Utilization

$$C_{n+1} = C_{n-1} * (1 - \alpha) + C_n * \alpha \tag{1}$$

where C_{n-1} is the last CPU utilization, C_n is the current CPU utilization, C_{n+1} is the predicted CPU utilization, and α is a configurable parameter between 0 and 1.

We use a simple history-based prediction model [8] widely used in literature as Eq. 1 to get the predicted CPU utilization in a node. Next, we use Eq. 2 to compare the current CPU utilization and the predicted CPU utilization in order to diagnose whether CPU utilization of a node is growing up. We implement the prediction model of CPU utilization trend in the runtime system for Map tasks and Reduce tasks respectively at each Slave node in a cloud, so each task can know whether it should continue running or temporarily yield the use of CPU resources for now.

Equation to Get CPU Utilization Trend

$$T(C_n,\ C_{n+1}) = \begin{cases} 1 & \text{if } C_n < C_{n+1} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where T() is the function of CPU utilization trend by returning 1 to indicate the growing of CPU utilization.

We use the algorithm in Fig. 2 to determine whether a task should be run or suspended. We make each Map task and Reduce task respectively execute the algorithm in order to continue running or suspending itself. We are sure that not all Map or Reduce tasks will be run or suspended simultaneously, because the algorithm will be executed only at a checkpoint function in Map and Reduce functions and CPU utilization in a node will be variant at run time.

```
function checkpoint():

if the current CPU utilization is differet to the last CPU utilization
then {
    if the prediction model of CPU utilization trend is not positive,
    then changing the status of the task
    if the task should be suspended,
    then suspending the task
}
```

Fig. 2. Algorithm in checkpoint function

4 Experiments

4.1 Implementation

We prepare 9 identical computers and each of them has an AMD Athlon II X4 620 CPU, 4 GB DDR2 RAM, 1 TB HD, and 1 Gbit Ethernet adapter. We use Gigabit Ethernet to connect the computers to each other and install Windows Server 2003 in all of them. We construct a cluster composed of the computers where one computer is used as a Master node and the other computers are used as Slave nodes.

We implement a tiny MapReduce system in order to observe the ASST performance. We do not implement ASST in Hadoop because Hadoop has other built-in components such as distributed file system (i.e. HDFS) [6] and distributed database (i.e. HBase) [9] to prevent us from clearly observing the ASST performance and because Hadoop currently cannot provide a task with CPU utilization information required by ASST. We implement the MapReduce system in the C language and make it support various essential functions, e.g., reading input files from a Master node, finding a suitable node to run a Map task, providing the Map task with input data, finding a suitable node to run a Reduce task, forwarding intermediate data from a node to another one, and collecting results from all nodes. Besides, we implement the MapReduce system to support the pipeline MapReduce enhancement [5] that moves intermediate data from Map tasks to Reduce tasks early instead of doing the movement in a specific shuffle phase. For suspending a task temporarily, we use API SwitchToThread to yield execution to another thread that is ready to run on the current processor, so we do not need to bring it back to execution later with an explicit API.

In the MapReduce system, we follow the working principle of Hadoop to implement a similar scheduler for comparison and briefly refer to it as Hadoop in the following experiments. In the following experiments, we observe the ASST performance in GREP [1], Inverted Index [10], Radix Sort [11], and Word Count [1]. In the prediction model of CPU utilization trend, we configure α in Eq. 1 to 0.5, so the last CPU utilization and the current CPU utilization of a node both have 50 % influence on the predicted CPU utilization. Meanwhile, we deploy input files in a Master node and let it distribute them over Slave nodes in the cluster at run time for various applications.

4.2 GREP

We program GREP [1] to search input files for a specific string. We develop a Map function to emit a file name as a key and a combination of a line number and the offset in the line number in where the string appears as a value in intermediate data. In GREP, we program a Map function to do most works but a Reduce function to merge outputs produced by a Map function. We sophisticatedly design GREP to find a rare string in input files, so a Map function will only produce very few intermediate data about 0.25 % input data. We input GREP with 128 files having 64 MB in each of them and get results in Fig. 3.

According to Fig. 3, we can observe that GREP work well when maximal Map and Reduce task numbers are less than 2. Because each of our computers has a quad-core CPU, we observe that GREP degrades performances when each node totally has more than 4 Map and Reduce tasks assigned by a Master node. Because a Map task does

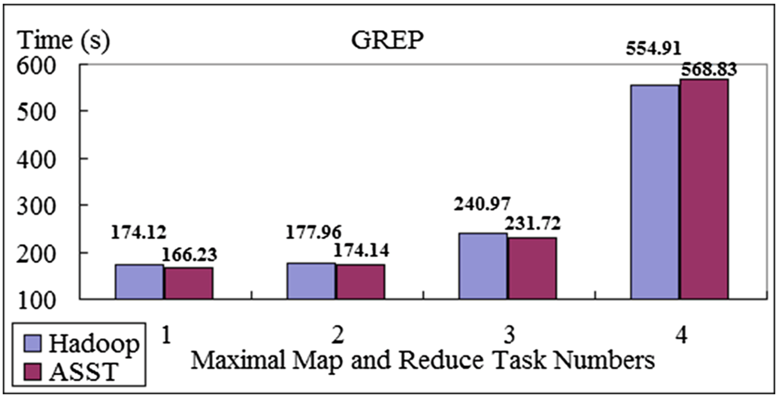


Fig. 3. GREP performance

most works, we observe that ASST does not work better than Hadoop by suspending certain tasks on demand when the node is overloaded by 4 Map tasks and 4 Reduce tasks. However, we show that the inappropriate configuration of slot numbers indeed has negative impacts on performances by speeding down the Hadoop performance to 3.19 (i.e. 3.19 = 554.91/174.12) and the ASST performance to 3.42 (i.e. 3.42 = 568.83/166.23) in the worst case. Nevertheless, we observe that ASST still outperforms Hadoop in most cases.

4.3 Inverted Index

We use Inverted Index [10] to get a summary of links appearing in input files. We program a Map function to emit a file name as a key and a combination of a link name and the offset related to the file as a value in intermediate data. In a Reduce function, we merge intermediate data produced by Map tasks. Although we test Inverted Index with 128 files having 64 MB in each of them, we can expect that intermediate data in Inverted Index is much more than GREP. We get results in Fig. 4.

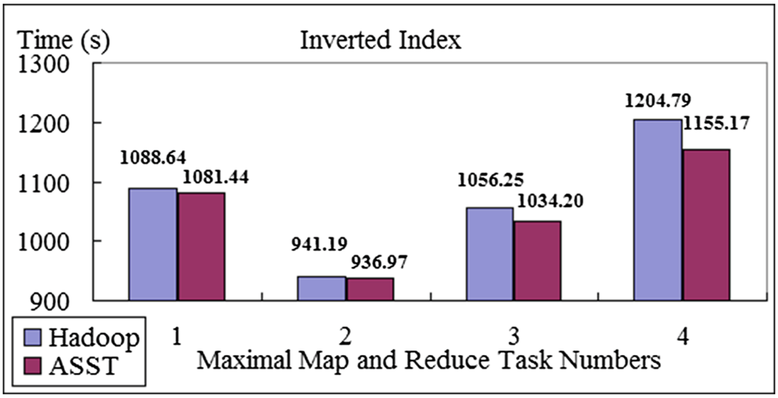


Fig. 4. Inverted index performance

In Fig. 4, we observe that the best configuration of the slot number in a node only approves a node to have 2 maximal Map tasks and 2 maximal Reduce tasks. When a node is only allowed to have 1 maximal Map task and 1 maximal Reduce task, we see that most CPU cores in nodes are idled. When approving a node to have more tasks, e.g. 4 maximal Map and Reduce tasks, we observe that the performance degrades seriously. Even so, we observe that ASST still outperforms Hadoop about 4.12 % (i.e. 4.12 % = (1204.79 − 1155.17)/1204.79). According to Fig. 4, we observe that ASST outperforms Hadoop in all configurations of slot numbers.

4.4 Radix Sort

We use Radix Sort [11] to sort numbers in input files. We program a Map function to scan input files for all numbers separated by space characters. However, we program the Map function to emit each word as a key but an empty string a value as intermediate data, so Radix Sort roughly has the quantity of intermediate data similar to that of input files. In a Reduce function, we sort and merge intermediate data produced by a Map function. Accordingly, we can expect that Radix Sort does most sorting works in Reduce tasks and has intermediate data roughly identical to input data. We give Radix Sort 128 files having 16 MB in each of them because sorting works cost much CPU time.

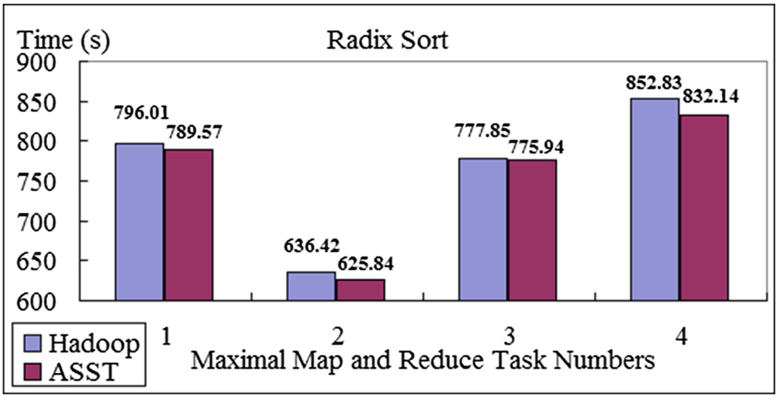


Fig. 5. Radix sort performance

In Fig. 5, we observe that Radix Sort still has the best performance when a node is approved to have 1 maximal Map task and 1 maximal Reduce task. At that time, we see that Radix Sort can use 636.42 s to finish the job in Hadoop but 625.84 s in ASST. Because Radix Sort costs many CPU computations in Reduce tasks, we observe that the performance suddenly degrades due to the inappropriate configuration of the slot number in a node. According to Fig. 5, we witness that ASST still outperform Hadoop in various configurations of slot numbers in nodes, because ASST can suspend certain Map tasks to avoid both producing too much intermediate data and competing with Reduce tasks for CPU resources. When the performance is degraded due to the inappropriate configuration of the slot number in a node, we see that ASST still can get a performance better than Hadoop.

4.5 Word Count

We use Word Count [1] to test ASST and compare it with Hadoop, because Word Count is a typical application widely used to test a MapReduce system on cloud computing. We program a Map function to scan input files for each word separated by space characters, and emit a word as a key and a string "1" as a value in intermediate data. In a Reduce function, we merge intermediate data produced by a Map function. We give Word Count 128 input files having 64 MB in each of them and show performances in Fig. 6.

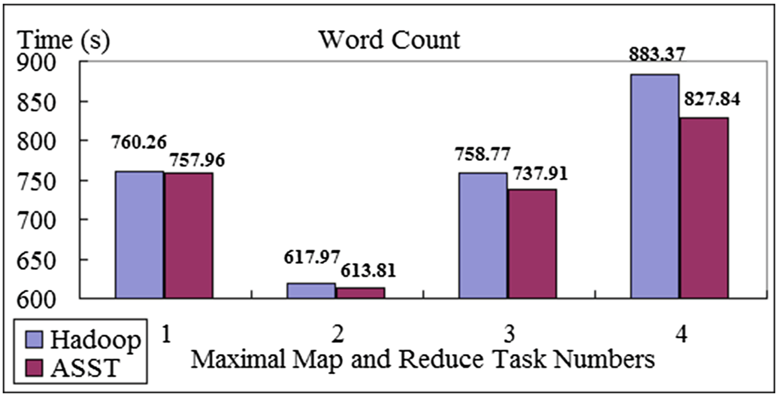


Fig. 6. Word Count performance

According to Fig. 6, we observe that the best configuration of the slot number in a node is 2 in the MapReduce system because Word Count can use 617.97 s in Hadoop and 613.81 s in ASST to finish their works. When a node is allowed to have more tasks by the inappropriate configuration of the slot number, we observe that the performance of Word Count is degraded in proportional to the allowed maximal task numbers. When a node is allowed to have 4 maximal Map tasks and 4 maximal Reduce tasks, for example, we observe that ASST still can outperform Hadoop about 6.29 % (i.e. 6.29 % = (883.37 − 827.84)/883.37). Because ASST can keep running certain tasks but suspend some tasks according to their types (i.e. Map tasks or Reduce tasks) on demand at run time, we observe that ASST is better than Hadoop in various cases.

5 Conclusions

In this paper, the Automatic Self-Suspended Task (ASST) for a MapReduce system is proposed to alleviate the performance degradation stemming from the inappropriate configuration of the slot number in a node on cloud computing. ASST can work for a MapReduce system and maintains a good performance regardless of how the slot number is configured in a node on cloud computing. ASST applies its working principle to Map and Reduce tasks respectively by continuously running or suspending certain of the tasks on demand. ASST uses the prediction model of CPU utilization trend to predict whether the CPU utilization in a node will go up or down in the future, and then respectively determines whether a Map task or a Reduce task should be run or suspended. ASST switches the status of a task on demand according to the prediction model of CPU utilization trend. ASST can be applied to a cloud where Map and Reduce tasks are executed simultaneously to get the pipeline MapReduce benefits or tasks of multiple applications are executed simultaneously to improve resource utilization. ASST is implemented in a tiny MapReduce system that supports the pipeline MapReduce enhancement, uses a task management scheduler similar to Hadoop, and has supports various functions available in most of the existing MapReduce systems. Besides, ASST is tested with GREP, Inverted Index, Radix Sort, and Word Count in experiments and compared with the control group having a configuration similar to Hadoop.

In the GREP experiment, we observe that the best configuration of the slot number is less than 2 because the performance begins to suddenly degrade in proportional to the increase of the slot number. Except for the case of approving a node to have 4 maximal Map and Reduce tasks, we observe that ASST outperforms Hadoop in all configurations of slot numbers in nodes. In the Inverted Index experiment, we observe that Inverted Index can get the best performance when a node is allowed to have 2 maximal Map and Reduce tasks. In each case of Inverted Index, we see that ASST can outperform Hadoop, even though the performance degrades due to the inappropriate configuration of the slot number in a node. For example, we observe that ASST still outperforms Hadoop about 4.12 %, when a node is allowed to have 4 maximal Map and Reduce tasks. In the Radix Sort experiment, we observe that performances of Hadoop and ASST both get the best performance with 2 maximal Map and Reduce tasks. Because Radix Sort costs many CPU computations in Reduce tasks, we observe that the performance suddenly degrades due to the inappropriate configuration of the slot number in a node. However, we see that ASST still outperforms Hadoop in various configurations of slot numbers in nodes. In the Word Count experiment, we observe that 2 maximal Map and Reduce tasks in each node are still the best configuration to performances. Nevertheless, we see that ASST still outperforms Hadoop. When a node is approved to have 4 maximal Map tasks and 4 maximal Reduce tasks, for example, we observe that ASST outperforms Hadoop about 6.29 %. Because ASST can keep running certain tasks but suspend some tasks temporarily according to their types (i.e. Map tasks or Reduce tasks) on demand at run time, we observe that ASST is better than Hadoop in various cases. Accordingly, we are convinced that ASST indeed can make a positive impact on the performance of a MapReduce system.

Acknowledgement. We thank the National Science Council of Taiwan for their support of this project under grant number NSC 102-2221-E-262-014 and NSC 101-2221-E-006-147-MY3. We thank Lunghwa University of Science and Technology for providing us with devices. We thank our friends in National Cheng Kung University for giving us some technical supports. We further offer our special thanks to the reviewers for their valuable comments and suggestions.

References

1. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)
2. Pallis, G.: Cloud computing: the new frontier of internet computing. IEEE Internet Comput. **14**(5), 70–73 (2010)
3. Murugesan, S.: Cloud computing: the new normal? Computer **46**(1), 77–79 (2013)
4. Brown, R.A.: Hadoop at home: large-scale computing at a small college. ACM SIGCSE Bull. **41**(1), 106–110 (2009)
5. Condie, T., Conway, N., Alvaro, P., Hellerstein, J.M., Elmeleegy, K., Sears, R.: MapReduce online. In: Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation, pp. 21–21 (2010)
6. Shvachko, K., Kuang, H., Radia, S., Chansler, R.: The Hadoop distributed file system. In: Proceedings of IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), pp. 1–10 (2010)
7. Silberschatz, A., Galvin, P.B., Gagne, G.: Operating System Concepts. Wiley, New York (2008). ISBN 978-0-470-12872-5
8. Everette, S., Gardner, J.: Exponential smoothing: the state of the art. J. Forecast. **4**(1), 1–28 (1985)
9. George, L.: HBase the Definitive Guide. O'Reilly Media, Sebastopol (2011). ISBN 978-1449396107
10. Mahapatra, A.K., Biswas, S.: Inverted indexes: types and techniques. IJCSI Int. J. Comput. Sci. Issues **8**(4), 384–392 (2011)
11. Davis, I.J.: A fast radix sort. Comput. J. **35**(6), 636–642 (1992)

Tag Completion and Refinement for Web Service via Low-Rank Matrix Completion

Lei Chen[1](✉), Geng Yang[2], Zhengyu Chen[1,3], Fu Xiao[1], and Xuxia Li[1]

[1] School of Computer, Nanjing University of Posts and Telecommunications, Nanjing, China
{chenlei,xiaof,lixx}@njupt.edu.cn, zych@jit.edu.cn

[2] Key Laboratory of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Nanjing, China
yangg@njupt.edu.cn

[3] School of Information Technology, Jinling Institute of Technology, Nanjing, China

Abstract. With the development of cloud computing, more and more web services are deployed on the cloud platform. It provides more solutions to the customers, but accompanies with a critical and fundamental problem, that is, how to easily find the desired web services. Since tags provide meaningful descriptions for web services function and non-function properties, some researchers have employed tags to facilitate web service discovery. However, the existing web service tags are often imprecise and incomplete. To complete the missing tags and correct the noisy ones, an efficient web service Tag Completion and Refinement based on Matrix Completion (TagCRMC) approach is proposed. The TagCRMC approach not only models the low-rank property of service-tag matrix, but also simultaneously integrates the content correlation consistency and the tag correlation consistency to ensure the correct correspondence between web services and tags. Experimental results on the real-world web services collection show the encouraging performance of the TagCRMC approach.

Keywords: Cloud computing · Web service · Tag completion · Tag refinement · Matrix completion

1 Introduction

In recent years, web service has become an important paradigm for developing web applications. Especially the emergence of Cloud infrastructure offers a powerful and economical platform to greatly facilitate the development and deployment of a large number of web services. With more and more web services being published on the cloud platform, a critical and fundamental problem is how to make web service easily discoverable to customers. Although WSDL (Web Service Description Language) documents can be utilized for web service discovery, limited information results in the low discovery performance and so impedes the widespread usage of web services.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 269–278, 2014.
DOI: 10.1007/978-3-319-13186-3_26

Since tags provide meaningful descriptions of objects, and allow users to organize and index their contents, tagging has become a popular mean to annotate various web resources, e.g., web image, online document, multimedia, and so on [1–3]. In web service domain, some real-world web service search engines, such as *SeekDa!*, have allowed users to manually annotate web services using tags. Web service tags are being treated as collective user knowledge to facilitate web service application, and attract a lot of attention. Some research works have been conducted to employ tags for web service discovery [4, 5]. For example, Ding et al. propose to improve the web service discovery performance by introducing tag data [4]. Fernandez et al. propose a mixed service discovery model based on service-tag relationships [5]. In Reference [5], users are encourage to provide tags to each web service to form tag-cloud, which could be matched using standard similarity measures against user requests. Then existing service tag-clouds are hierarchically clustered to achieve lightweight, browsable service ontologies, represented by discriminating tags per cluster.

A common premise of these works is that the annotated tags are known, accurate and complete. However, this premise may not be true. In practice, many web service tags provided by customers are imprecise and incomplete [6–8]. There are two main reasons concerning this fact [8]: (1) users are not always willing to submit tags and so the number of tags that they enter is usually small, in other words, the tags are usually missing; (2) web service tags may also be annotated by attackers, or users may select different words for expressing the same concept or the same word for different concept, which creates some noisy tags. These inevitable noisy and missing tags make it harder to find service tagged by other users.

To handle this problem, Chen et al. propose to recommend tags to web services with few tags according to the tag co-occurrence [6, 7]. Katakis et al. propose to automate tagging services by modeling this problem as a multi-label classification problem [8]. Azmeh et al. propose to employ machine learning technology and WordNet synsets to automatically annotate tags to web service [9]. Fang et al. also propose a automatic web service tagging approach based on tag enriching and tag extraction [10]. In Reference [10], web services are first clustered according to WSDL documents, and the enriched tags for a service are the tags of other web service in the same cluster, and then recommended tags are extracted from WSDL documents and related descriptions. Despite these works have shown encouraging results in their respective dataset, they requires a large clean tags as training set to train a reliable tag recommendation model, and any noisy tags could potentially lead to a significant prediction bias.

In this paper, in order to complete the missing tags, and simultaneously correct the noisy ones, we propose a novel web service Tag Completion and Refinement based on Matrix Completion (TagCRMC) approach. Firstly, we formulate web service tag completion and refinement problem as a Modified Matrix Completion (MMC) problem. Secondly, in order to efficiently solve the MMC problem, we also present an Operator Splitting based Iterative Threshold (OSIT) algorithm. The TagCRMC approach is motivated by the following three main facts: (1) low-rank. The existing work on text information processing [11] has demonstrated that the semantic space spanned by text keywords can be approximated by a smaller subset of salient words derived from the original space. As one kind of text information, service tags are consequently subject to such low-rank property; (2) Content correlation consistency. By analyzing the large-scale WSDL documents and the corresponding service tags, we can observe that those web services with the same or similar function are often described by the similar WSDL documents and thus are typically annotated with similar tags. Content correlation consistency describes the relationships between WSDL document level and tag annotation level; (3) Tag correlation consistency. Tags associated with web services do not appear in isolation, instead often appear correlatively and naturally interact with each other at the semantic level. As another important prior knowledge, tag correlation characterizes the relationships within semantic level and is often the preliminary assumption of multi-label and contextual learning algorithms [12].

The main contributions of this paper are summarized as follows:

(1) We propose a TagCRMC approach to complete the missing tags and correct the noisy ones by formulating web service tag completion and refinement problem as a MMC problem, which derived from the conventional matrix completion problem, and naturally inherits the advantages of Matrix Completion on processing the large-scale (approximately) low-rank data.

(2) The proposed TagCRMC approach not only models the low-rank property of service-tag matrix, but also integrates the content correlation consistency and the tag correlation consistency in a simultaneously and seamless manner, which ensures the correct correspondences between web services and tags.

(3) We employ the Operator Splitting technique to design an efficient OSIT algorithm to solve the proposed MMC problem, and the final experimental results demonstrate its encouraging performance.

The rest of this paper is organized as follows. Section 2 describes the TagCRMC approach in detail by formulating web service tag completion and refinement problem as a MMC problem. Section 3 presents an efficient OSIT algorithm to solve the proposed MMC problem. Section 4 demonstrates the feasibility of the proposed TagCRMC approach using real-world web services collection. Finally, we draw conclusions and discuss future work in Sect. 5.

2 Web Service Tag Completion and Refinement Based on Matrix Completion (TagCRMC)

In this Section, we propose the TagCRMC approach to complete the missing tags and correct the noisy ones by formulating the web service tag completion and refinement problem as a MMC problem.

Supposed m and n be the number of web services and tags, respectively. Then the initial incomplete and noisy service-tag matrix $M_{ST} \in \mathbb{R}^{m \times n}$ can be derived from user-provided tags, where $M_{ST}(i, j)$ is set to 1 if tag t_j is assigned to service ws_i and 0 otherwise. In addition, following Reference [13], the tag similarity matrix $M_{TT} \in \mathbb{R}^{n \times n}$ can be estimated by counting the tags occurrence frequency and co-occurrence frequency, that is, we may define the tag similarity between two tags t_i and t_j as follows:

$$M_{TT}(i,j) = n(t_i, t_j)/(n(t_i) + n(t_j) - n(t_i, t_j)) \tag{1}$$

where $n(t_i)$ and $n(t_j)$ are the occurrence of tags t_i and t_j, and $n(t_i, t_j)$ is the co-occurrence of tags t_i and t_j. In addition, the service-feature matrix $M_{SF} \in \mathbb{R}^{m \times l}$ can be naturally obtained by processing web service information such as WSDL documents, service name and service description (the detailed explanation about the feature extracting process for web service can refer to our prior work [14]). Finally, we denote by $X \in \mathbb{R}^{m \times n}$ the final service-tag matrix that needs to be computed, where each entry in $X(i, j)$ represents the confidence score for a tag t_j associated with a web service ws_i. Therefore, when reconstructing the final service-tag matrix X based on the initial service-tag matrix M_{ST}, the tag similarity matrix M_{TT} and the service-feature matrix M_{SF}, we can formulate it as the following MMC problem:

$$\min_{X \in R^{m \times n}} \left\{ rank(X) + \frac{\lambda}{2}[C_c(X) + C_t(X)] \quad \text{s.t. } \|\mathrm{P}_\Omega(X - M_{ST})\|_F^2 \le \delta \right\} \tag{2}$$

where $C_c(X)$ and $C_t(X)$ are used to measure the content correlation consistency and the tag correlation consistency, and $\Omega = \{(i,j)|1 \le i \le m, 1 \le j \le n, M_{ST}(i,j) = 1\}$, $\mathrm{P}_\Omega(\cdot)$ is the projector operator defined as:

$$[\mathrm{P}_\Omega(X)]_{ij} = \begin{cases} X(i,j) & if\ (i,j) \in \Omega, \\ 0 & otherwise. \end{cases} \tag{3}$$

For the MMC problem, the content correlation consistency constraint $C_c(X)$ and tag correlation consistency constraint $C_t(X)$ can be formulated as follows.

Firstly, the final service-tag matrix X should reflect the textual content of web services represented by the service-feature matrix M_{SF}, where each web service ws_i is represented as a row vector $M_{SF}(i,:)$. To satisfy the content correlation consistence, we propose exploiting this constraint by comparing web service similarity based on textual content with web service similarity based on the overlap in annotated tags. More specifically, we compute the textual similarity between web services ws_i and ws_j as $M_{SF}(i,:)M_{SF}(j,:)^{\mathrm{T}}$. On the other hand, for the final service-tag matrix X, we can also compute the service similarity between web services ws_i and ws_j based on the overlap between their tags, i.e., $X(i,:)X(j,:)^{\mathrm{T}}$. If the final service-tag matrix X reflects the textual content of web services, we expect $\left|X(i,:)X(j,:)^{\mathrm{T}} - M_{SF}(i,:)M_{SF}(j,:)^{\mathrm{T}}\right|^2$ to be small for any two web service ws_i and ws_j. As a result, in order to satisfy the content correlation consistence constraint, we have that

$$C_c(X) = \sum_i \sum_j \left| X(i,:)X(j,:)^{\mathrm{T}} - M_{SF}(i,:)M_{SF}(j,:)^{\mathrm{T}} \right|^2 = \left\| XX^{\mathrm{T}} - M_{SF}M_{SF}^{\mathrm{T}} \right\|_F^2 \tag{4}$$

and expect a small value for $C_c(X)$.

Secondly, similar to the content correlation consistency constraint, we also expect the final service-tag matrix X to be consistent with the tag similarity matrix M_{TT}. Therefore, we have that

$$C_t(X) = \left\| X^{\mathrm{T}}X - M_{TT} \right\|_F^2 \tag{5}$$

and expect a small value for $C_t(X)$.

Incorporating the Eqs. (4) and (5) into Eq. (2), we have the following MMC problem for web service tag completion and refinement:

$$\min_{X \in R^{m \times n}} \left\{ \begin{array}{l} rank(X) + \frac{\lambda}{2}\left[\left\| XX^{\mathrm{T}} - M_{SF}M_{SF}^{\mathrm{T}} \right\|_F^2 + \left\| X^{\mathrm{T}}X - M_{TT} \right\|_F^2 \right] \\ \text{s.t. } \left\| \mathrm{P}_{\Omega}(X - M_{ST}) \right\|_F^2 \le \delta \end{array} \right\} \tag{6}$$

Unfortunately, *rank*(X) is a non-convex function and difficult to optimize. Following recent work in matrix completion [15, 16], we relax *rank*(X) with the convex *nuclear*-norm $\|X\|_*$, then we can reformulate the MMC problem (6) as follows:

$$\min_{X \in R^{m \times n}} \mu \|X\|_* + \frac{1}{2}\mu\lambda \left(\left\| XX^{\mathrm{T}} - M_{SF}M_{SF}^{\mathrm{T}} \right\|_F^2 + \left\| X^{\mathrm{T}}X - M_{TT} \right\|_F^2 \right) + \frac{1}{2} \left\| \mathrm{P}_{\Omega}(X - M_{ST}) \right\|_F^2 \tag{7}$$

3 Operator Splitting Based Iterative Threshold (OSIT) Algorithm for the MMC Problem

In this Section, we propose an efficient Operator Splitting based Iterative Threshold (OSIT) algorithm to solve the MMC problem in Eq. (7). Compared to the other optimization approaches such as Newton's method and interior point methods [17], the OSIT algorithm is advantageous in that its computational complexity per iteration is significantly lower, making it suitable for large-scale datasets.

Next, in order to better introduce the OSIT algorithm, we first present some definitions and theorems, and then describe the detailed steps for OSIT algorithm.

Definition 1. Singular value shrinkage operator. Suppose the Singular Value Decomposition (SVD) of a matrix $X \in \mathbb{R}^{m \times n}$ with rank r:

$$X = U\Lambda V^{\mathrm{T}} \tag{8}$$

where U and V are respectively $m \times r$ and $n \times r$ matrices with orthonormal columns, $\Lambda = diag(\{\sigma_i | 1 \leq i \leq r\})$, and the singular values σ_i are positive. For each $\tau \geq 0$, we define the singular shrinkage operator D_τ as follows:

$$D_\tau(X) = US_\tau(\Lambda)V^{\mathrm{T}} \tag{9}$$

where $S_\tau(\Lambda) = diag(\{\sigma_i - \tau\}_+)$, $t_+ = \max(0, t)$.

Theorem 1. For any $\tau > 0$ and $Y \in \mathbb{R}^{m \times n}$, the singular value shrinkage operator obeys:

$$D_\tau(Y) = \arg \min_{X \in R^{m \times n}} \left\{ \tau \|X\|_* + \frac{1}{2} \|X - Y\|_F^2 \right\} \tag{10}$$

Proof. The proof of Theorem 1 follows similar lines of the proof in Reference [15].

Theorem 2. Given the following unconstrained convex problem

$$\min_{X \in \aleph} F(X) = F_1(X) + F_2(X) \tag{11}$$

where $\aleph$ is a Hilbert space, and $F_1(X)$ is a convex, smooth and lower semi-continuous function, $F_2(X)$ is a convex, smooth and Lipschitz continuous function. Then, the following iterative sequence (12) will converges to a minimizer of Eq. (11):

$$X_{k+1} = \arg \min_{X \in \aleph} \tau F_1(X) + \frac{1}{2} \|X - (X_k - \tau \nabla F_2(X_k))\|_F^2 \tag{12}$$

where τ is a non-negative real number.

Proof. The proof of Theorem 2 follows similar lines of the proof in Refence [18].

Our OSIT algorithm is inspired by the Operator Splitting technique in Theorem 2. Observing the MMC problem in Eq. (7), we can have that $\mu \|X\|_*$ and $\frac{1}{2} \|\mathrm{P}_\Omega(X - M_{ST})\|_F^2$ is convex, and $\frac{1}{2} \mu\lambda (\|XX^{\mathrm{T}} - M_{SF} M_{SF}^{\mathrm{T}}\|_F^2 + \|X^{\mathrm{T}}X - M_{TT}\|_F^2)$ is also convex. Consequently, the linear combination of convex terms, namely the MMC problem (7) is convex and it has at least one minimizer. Based on the Eqs. (7) and (11), we can have the function $F(X) = F_1(X) + F_2(X)$, where$F_1(X) = \mu \|X\|_*$ $F_2(X) = \frac{1}{2} (\mu\lambda \|XX^{\mathrm{T}} - M_{SF} M_{SF}^{\mathrm{T}}\|_F^2 + \mu\lambda \|X^{\mathrm{T}}X - M_{TT}\|_F^2 + \|\mathrm{P}_\Omega(X - M_{ST})\|_F^2)$. Furthermore, we have

$$\nabla F_2(X) = 2\mu\lambda(XX^{\mathrm{T}} - M_{SF} M_{SF}^{\mathrm{T}})X + 2\mu\lambda X(X^{\mathrm{T}}X - M_{TT}) + \mathrm{P}_\Omega(X - M_{ST}) \tag{13}$$

Based on the aforementioned analysis and deduction, we summarize the Operator Splitting based Iterative Threshold (OSIT) algorithm in Algorithm 1.

Algorithm 1: Operator Splitting based Iterative Threshold (OSIT) algorithm

Input: the initial observed service-tag matrix $M_{ST} \in \mathbb{R}^{m \times n}$, the service-feature matrix $M_{SF} \in \mathbb{R}^{m \times l}$, the tag similarity matrix $M_{TT} \in \mathbb{R}^{n \times n}$, parameter λ, μ and τ.

Output: The final service-tag matrix $X_{ST} \in \mathbb{R}^{m \times n}$.

```
Compute the projector operator $\mathrm{P}_\Omega(\cdot)$ according to the initial service-tag matrix $M_{ST} \in \mathbb{R}^{m \times n}$;
Initialize $k$=0, $X_0 = \mathrm{P}_\Omega(M_{ST})$;
While not converged do
    $Y_k = X_k - \tau(2\mu\lambda((X_k X_k^{\mathrm{T}} - M_{SF} M_{SF}^{\mathrm{T}}) X_k + X_k (X_k^{\mathrm{T}} X_k - M_{TT})) + \mathrm{P}_\Omega(X_k - M_{ST}))$;
    $(U, \Sigma, V) = svd(Y_k)$;
    $X_{k+1} = U S_{\tau\mu}(\Sigma) V^{\mathrm{T}}$;
    $k$=$k$+1;
End while;
Output $X_{ST} \leftarrow X_{k+1}$.
```

The convergence rate for the OSIT algorithm is $O(1/\sqrt{k})$, where k is the number of iterations, and which is guaranteed by the convergence property of the Operator Splitting technique [18]. The space requirement for the OSIT algorithm is $O(m \times n)$, where m is the number of web services and n is the number of tags.

4 Experimental Evaluations

In this Section, we first give a brief description of experimental dataset, and then present the experimental results on web service tag completion and refinement.

4.1 Experimental Dataset

To verify the performance of web service tag completion and refinement, we select the publicly published STag 1.0 dataset from http://www.zjujason.com/ as our experimental dataset. This dataset contains 15,968 web services crawled from web service search engine *Seekda!*, for each record of web service, it has service name, service description, tags, and WSDL document. Note that the tags in this dataset are rather noisy and some of them are misspelling or meaningless words. Hence, a pre-processing was performed to filter out these tags. We matched each tag with entries in WordNet (http://wordnet.princeton.edu/) and only the tags with coordinates in WordNet were retained. Since the manual creation of ground truth is an expensive process, here we only select 200 web services which contain 117 unique tags as the evaluation dataset, and its ground truth is manually created by the volunteers with more than 3 years experiences in developing web services.

4.2 Performance Evaluation of Tag Completion and Refinement

To evaluate the performance of web service tag completion and refinement, the F-score, which was widely used as evaluation metric of tag completion and refinement [3, 19], was calculated to measure the completion and refinement results for each tag and average them as the final evaluation. The F-score of tag t_i and Ave-F-score are defined as:

$$\text{F-score}(t_i) = \frac{2 \times precision(t_i) \times recall(t_i)}{precision(t_i) + recall(t_i)} \tag{14}$$

$$\text{Ave-F-score} = \sum\nolimits_{i=1}^{n} \text{F-score}(t_i) \Big/ n \tag{15}$$

where $recall(t_i)$ is defined as the number of web services correctly annotated with t_i divided by the number of web services that have t_i in the ground truth. $precision(t_i)$ is defined as the number of correctly annotated web services divided by the total number of web services annotated with t_i.

In order to compare the tag completion and refinement performance, the following three algorithms as well as the original tagging information are employed as the baselines:

- Original Tag Information (OTI): the original user-provided tags information.
- Tag RecomMendation (TRM): tag refinement algorithm based on tag co-occurrence in Reference [7].
- Tag Annotating Automatically (TAA): tag refinement algorithm using machine learning and WordNet synsets in Reference [9].
- Tag Enriching and Extraction (TEE): tag refinement algorithm based on tag enriching and tag extraction in Reference [10].

In our experiments, we first employ our previous proposed Wordnet-powered Supervised Web Service Representation algorithm [14] to extracted web services as a 200 × 300 service-feature matrix. At the same time, we set $\mu = 5$, $\lambda = 2$, $\tau = 0.001$ by cross validation. In our OSIT algorithm, the low rank matrix X_{ST} is the final refined result, where each entry produces the confidence score for a tag associated with a web service. To identify the ultimate web service tags, we set a threshold $\theta = 0.45$ by cross-validation to generate the final tags associated with a given web service, that is, if $X_{ST}(i,j) \geq \theta$, then tag t_j is assigned to web service ws_i. Table 1 shows the average performance of different algorithm for all 117 tags. Form Table 1, we can observed that our TagCRMC algorithm outperformed the four state-of-the-art algorithms.

Table 1. Average performance of different algorithms for tag completion and refinement

Algorithm	TagCRMC	OTI	TRM	TAA	TEE
Ave-F-score	0.719	0.468	0.707	0.659	0.676

In addition, we evaluate the computational efficiency by the running time for web service tag completion and refinement. All the experiments are implemented with JDK

1.7.0-10, Eclipse 3.6.0. They are conducted on a computer with Intel(R) Core(TM) i7-3520 M CPU @ 2.90 GHz with 4 GB of RAM, running Windows 7 OS. Note that for the OTI method, the original user-provided tags information is directly adopted without taking any running time, so we only report the running time for the other four methods. Table 2 summarizes the running times of both the proposed TagCRMC and the three baseline methods for tag completion and refinement. From Table 2, we can observe that the proposed TagCRMC takes more running time, which mainly due to the use of full SVD. However, from the Algorithm 1, we can see that computing the full SVD for the OSIT is unnecessary: we only need those singular values that are larger than a threshold '$\tau\mu$' and their corresponding singular vectors. So a software package PROPACK [20] can be employed to efficiently alleviate the TagCRMC's running time by computing the partial SVD.

Table 2. Running times (seconds) of different algorithms for tag completion and refinement

Algorithm	TagCRMC	TRM	TAA	TEE
Time	11.23	2.92	4.35	5.47

5 Conclusion and Future Work

Motivated by the fact that the existing user-provided service tags in public search engine are imprecise and incomplete, we proposed an efficient TagCRMC approach for web service tag completion and refinement. Experimental results on real-world web service collection well demonstrated the effectiveness of our proposed approach.

In the future, we will expand the scale of experimental dataset by inviting more volunteers for further experimental evaluation. In addition, we plan to exploit more efficient iterative optimization algorithm to solve the proposed MMC problem, so than the proposed TagCRMC approach can work on even larger volume services corpus.

Acknowledgements. This work is supported by the National Natural Science Foundation (Nos. 61272084, 61201163, 61272422 and 61373137), the National Key Basic Research Development Program (No. 2011CB302903), and the Natural Science Foundation of Jiangsu Province (Nos. BK2011754 and BK20130096), the Key University Natural Science Research Program of Jiangsu (No. 11KJA520002), the Research Fund for the Doctoral Program of High Education (Nos. 20113223110003 and 20093223120001).

References

1. Ames, M., Naaman, M.: Why we tag: motivations for annotation in mobile and online media. In: 2007 SIGCHI Conference on Human Factors in Computing Systems (CHI), pp. 971–980 (2007)
2. Sigurbj¨ornsson, B., Van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: 17th International Conference on World Wide Web (WWW), pp. 327–336 (2008)

3. Wu, L., Jin, R., Jain, A.K.: Tag completion for image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **35**(3), 716–727 (2013)
4. Ding, Z., Lei, D., Yan, J.: A Web service discovery method based on tag. In: 2010 International Conference on Complex, Intelligent and Software Intensive Systems, pp. 404–408 (2010)
5. Fernandez, A., Hayes, C., Loutas, N.: Closing the service discovery gap by collaborative tagging and clustering techniques. In: 2008 International Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web, pp. 115–128 (2008)
6. Chen, L., Zheng, Z., Feng, Y.: WSTRank: ranking tags to facilitate Web service mining. In: 10th International Conference on Service Oriented Computing, pp. 12–15 (2012)
7. Chen, L., Wang, Y., Yu, Q.: WT-LDA: user tagging augmented LDA for Web service clustering. In: 11th International Conference on Service Oriented Computing, Berlin, Germany, pp. 1–15 (2013)
8. Katakis, I., Pallis, G., Dikalakos M.: Automated tagging for the retrieval of software resources in grid and cloud infrastructures. In: 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 628–635 (2012)
9. Azmeh, Z., Falleri, J.-R., Huchard, M., Tibermacine, C.: Automatic Web service tagging using machine learning and WordNet synsets. In: Filipe, J., Cordeiro, J. (eds.) WEBIST 2010. LNBIP, vol. 75, pp. 46–59. Springer, Heidelberg (2011)
10. Fang, L., Wang, L., Li, M.: Towards automatic tagging for Web services. In: 10th IEEE International Conference on Web Services, pp. 528–535 (2012)
11. Zhao, R., Grosky, W.: Narrowing the semantic gap improved text-based Web document retrieval using visual features. IEEE Trans. Multimedia **4**(2), 189–200 (2002)
12. Zhu, G., Yan, S., MA, Y.: Image tag refinement towards low-rank, content-tag prior and error sparsity. In: 2010 ACM Multimedia, pp. 461–470 (2010)
13. Liu, J., Zhang, Y., Li, Z.: Correlation consistency constrained probabilistic matrix factorization for social tag refinement. Neurocomputing **119**, 3–9 (2013)
14. Chen, L., Yang, G., Zhu, W.: Clustering facilitated Web services discovery model based on supervised term weighting and adaptive metric learning. Int. J. Web Eng. Technol. **8**(1), 58–80 (2013)
15. Cai, J., Candes, E., Shen, Z.A.: Singular value thresholding algorithm for matrix completion. SIAM Journal of Optimization **20**(4), 1956–1982 (2010)
16. Ma, S., Goldfarb, D., Chen, L.: Fixed point and Bregman iterative methods for matrix rank minimization. Mathe. Program. Ser. A **128**(1), 321–353 (2011)
17. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2009)
18. Combettes, P., Wajs, V.: Signal recovery by proximal forward-backward splitting, multiscale modeling and simulation. SIAM Interdisc. J. **4**, 1168–1200 (2005)
19. Wang, C., Jing, F., Zhang, L.: Content-based image annotation refinement. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 123–130 (2007)
20. Larsen, R.M.: Lanczos Bidiagonalization with Partial Reorthogonalization. Aarhus University, Technical report, DAIMI PB-357, code (1998). http://soi.stanford.edu/~rmunk/PROPACK/

Mobile Sensing, Mining and Visualization for Human Behavior Inference

A Differential Approach for Identifying Important Student Learning Behavior Patterns with Evolving Usage over Time

John S. Kinnebrew[1](✉), Daniel L.C. Mack[2], Gautam Biswas[1], and Chih-Kai Chang[3]

[1] Department of EECS and ISIS, Vanderbilt University, Nashville, TN, USA
john.s.kinnebrew@vanderbilt.edu
[2] Baseball Analytics, Kansas City Royals, Kansas City, MO, USA
[3] Department of Information and Learning Technology,
National University of Tainan, Tainan, Taiwan

Abstract. Effective design and improvement of dynamic feedback in computer-based learning environments requires the ability to assess the effectiveness of a variety of feedback options, not only in terms of overall performance and learning, but also in terms of more subtle effects on students' learning behavior and understanding. In this paper, we present a novel interestingness measure, and corresponding data mining and visualization approach, which aids the investigation and understanding of students' learning behaviors. The presented approach identifies sequential patterns of activity that distinguish groups of students (e.g., groups that received different feedback during extended, complex learning activities) by differences in both total behavior pattern usage and evolution of pattern usage over time. We demonstrate the utility of this technique through application to student learning activity data from a recent experiment with the Betty's Brain learning environment and four different feedback and learning scaffolding conditions.

Keywords: Interestingness measure · Sequence mining · Learning behaviors · Information gain

1 Introduction

In order to more effectively teach and promote skills required in the modern world of near-ubiquitous computing and internet connectivity, computer-based learning environments have become more complex and open-ended. This complexity also drives a need for dynamic, adaptive feedback and learning scaffolding that can support students in understanding how to employ and learn with these environments and tools. However, in order to effectively design and improve such feedback/scaffolding, we must first be able to assess the effectiveness of a variety of scaffolding options, not only in terms of overall performance and learning, but also in terms of more subtle effects on students' behavior and understanding.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 281–292, 2014.
DOI: 10.1007/978-3-319-13186-3_27

Identifying sequential patterns in learning activity data has provided a useful tool for discovering and better understanding such student learning behaviors. However, once these behavior patterns are mined, researchers must interpret and analyze the resulting patterns to identify a relevant subset of important patterns that provide a basis for generating actionable insights about how students learn, solve problems, and interact with the environment. Algorithms for mining sequential patterns generally associate a measure of pattern frequency in the data with the relative importance or ranking of the pattern for investigation. However, when analyzing students' learning behaviors, another important aspect of these patterns is the evolution of their usage over the course of a student's learning or problem-solving activities. Further, in order to understand how different feedback and scaffolding can affect learning behavior, the more important patterns are those that differ across different experimental groups or other categories of students.

To address this challenge, we present a data mining and visualization approach, combining traditional sequence mining and a novel information-theoretic interestingness measure for ranking behavior patterns that distinguish groups of sequences (e.g., groups of students in different experimental conditions) by differences in both total pattern usage and the evolution of pattern usage over time. This approach builds on the Temporal Interestingness of Patterns in Sequences (TIPS) analysis framework [1], but instead of basing interestingness on the extent of changes in usage over time as in TIPS, this approach incorporates the dimension of student/sequence groups to identify and visualize patterns that are used differentially across those groups (by either total frequency or relative usage patterns over time). To demonstrate the utility of this approach, we present case study results of mining student activity data from a recent experiment with the Betty's Brain learning environment. These results illustrate the effectiveness of our approach and suggest further refinements for temporal and differential analysis of sequential learning activity data.

2 Related Work

The canonical sequential pattern mining task is to discover sequential patterns of items that are found in many of the sequences in a given dataset [2,3]. Researchers have applied variations of sequence mining techniques to a wide range of educational data in order to better understand and improve learning. For example, researchers have employed sequential pattern mining to better understand specific aspects of student learning behavior, such as Nesbit *et al.* ([4]), who mined the longest common subsequences in data from the gStudy learning environment to research how students self-regulate as they learn. Others have employed sequence mining techniques to more directly scaffold and improve student learning. For example, Perera *et al.* ([5]) help student groups collaborating on software development to improve their work by observing and emulating the behaviors of the strong groups, which are determined through sequence mining. Kinnebrew *et al.* [6,7] compared sequential patterns mined from student activity sequences to identify patterns for targeted feedback that differ in frequency between student groups and between productive and unproductive periods of work. Other researchers have also employed sequential pattern mining to identify differences among student groups or generate student models to customize learning to individual students [8–10].

Sequence mining algorithms commonly rank the discovered patterns by their frequency in the dataset. Over time, researchers have developed additional measures to utilize properties other than just frequency to rank mined patterns [11]. These measures are often referred to as "interestingness measures" and have been applied to a variety of data mining tasks, such as sequence mining and association rule mining [12]. The measures can be simple calculations like accuracy, specificity, and recall, or more complicated ones like the Gini Index [13] and information gain [14], which identify patterns that possess more complex relationships to the data. In educational data mining applications, interestingness measures have been used to rank mined association rules (e.g., [15]). More generally, information gain has been employed with educational data to solve problems like identifying attributes and producing rules for predicting student performance [16].

In addition to an interestingness measure to identify potentially important patterns, our approach also employs a visualization of pattern usage over time with heat maps to more efficiently analyze identified patterns. Heat maps have commonly been employed in biomedical fields for tasks including visualization of gene expression [17] and identification of common gene locations. Heat maps have also seen use in other fields to trace dynamic changes over time, such as in the environment or in music habits, and often employ maps that contain non-regular shapes, especially those of real-world geography [18].

3 Identifying Interesting Differences in Evolving Pattern Usage

With long sequences of temporal data, such as student learning activities in a computer-based learning environment, researchers and analysts are not only interested in discovering frequent sequential patterns, but, in many cases, also need to analyze their occurrence over time and identify patterns that differ among groups of sequences. To address these needs, we present the Differential, Temporal Interestingness of Patterns in Sequences (D-TIPS) measure and approach for identifying and visualizing patterns that are employed differentially over time across groups of students (e.g., groups that receive different scaffolding in a learning environment). The first step in analyzing learning activity sequences is to define and extract the actions that make up those sequences from interaction traces logged by the environment. The definition of actions in these sequences for Betty's Brain data is discussed further in Sect. 4. Given a set of sequences corresponding to the series of actions performed by each student, the D-TIPS technique consists of four primary steps that extend the corresponding steps in TIPS [1] to identify and interpret differences among student/sequence groups instead of simply over time:

1. Generate candidate patterns that are common to many students in one or more groups through sequential pattern mining.
2. Calculate a temporal footprint for each candidate pattern by mapping it back to locations where it occurs in the activity sequences.
3. Provide a ranking of the candidate patterns using an information-theoretic interestingness measure applied to the temporal footprint of each pattern across sequence groups.
4. For the highly-ranked, differential patterns, visualize their temporal footprints using heat maps to compare trends and spikes in usage across groups.

3.1 Identifying Common Patterns

In order to identify candidate behavior patterns for investigation, we employ a sequential pattern mining algorithm to the set of student activity sequences. Standard sequential pattern mining produces a set of sequential patterns that meet a given support threshold (i.e., they occur in at least a given percentage of the sequences). Certain types of additional constraints can be imposed with some algorithms, but otherwise the choice of algorithm does not affect the resulting set of patterns. In this case, because we are interested in behavior patterns where the actions occur immediately or shortly following each other in the sequence, we employ an algorithm (from Pex-SPAM [19]) that allows constraints on the gaps between actions in the identified sequential patterns[1]. Further, to identify patterns common to the majority of the students in any group, we apply sequential pattern mining to each group of student sequences separately with a support threshold of 50 % and combine the resulting sets of patterns to identify the full set of candidate patterns.

Once the common behavior patterns are mined, researchers must interpret and analyze the resulting patterns to identify a relevant subset of important patterns that provide a basis for generating actionable insights (*e.g.*, how to scaffold user interactions with the learning environment to encourage specific, productive behaviors). A pattern's "frequency" in sequential pattern mining terms (*i.e.*, the number of sequences in which the pattern occurs) is often used to rank the importance of the identified patterns. Alternatively, the frequency of occurrence within sequences is a different frequency measure that can be more appropriate for long sequences [6]. For learning activity sequences, this occurrence frequency (*i.e.*, how often a pattern occurs *within* sequences) is generally of more interest than simply the number of students who employed a pattern at least once. D-TIPS relies on a consideration of pattern occurrence within sequences, but rather than simply ranking patterns by their occurrence frequency (e.g., average or median occurrence frequency among the students/sequences), it also incorporates information about how this frequency changes over time.

[1] In the results presented in Sect. 5, we allowed a maximum gap of one action, allowing up to one irrelevant or variable action between consecutive actions in the pattern. However, in general, other sizes of gaps or no gap at all may be appropriate depending on the data and goals of an analysis.

3.2 Calculating the Temporal Footprint

Given a set of candidate patterns from the sequential pattern mining, the next step is to map the patterns back to the activity sequences to define a temporal footprint for each pattern. Each sequence is divided into n consecutive slices, such that each contains $\frac{100}{n}$ % of the student's actions in the full sequence. Corresponding slices (e.g., the first slice from each sequence, the second slice from each, and so on) are then grouped into bins to define the temporal footprint of the sequence. Although the slices for different student sequences can be of different lengths (and there may be different numbers of students in each group), the interestingness measure described in the next section is calculated with respect to all of the actions in each bin and takes into account the proportion of total actions falling into each bin.

The number of slices/bins chosen for the temporal footprint is a parameter that determines the level of temporal granularity for the analysis. In addition to the desired granularity, an important consideration for choosing the number of bins is the quantity and variability of the data available. In particular, finer granularities (i.e., more bins) increase the likelihood that the analysis will be overwhelmed by random variation and noise. For example, spikes and other differences in pattern frequency that are identified across bins with relatively small quantities of actions are more likely to be the result of noise than with larger bins. With Betty's Brain data, initial qualitative analysis has suggested that with relatively few activity sequences (e.g., 10 to 15 students), 3 to 5 bins may be most effective, while with more sequences (e.g., 30 or more students), anything from 5 to 10 bins tends to work well.

3.3 D-TIPS for Ranking Patterns by Interestingness

In order to identify more interesting patterns by their difference in temporal usage across groups, the D-TIPS interestingness measure applies information gain (IG) with respect to pattern occurrence across the groups in each of the n corresponding bins of their temporal footprints. Information gain is defined as the difference in expected information entropy [20] between one state and another state where some additional information is known (e.g., the difference between a set of data points considered as a homogeneous group versus one split into multiple groups based on the value of some other feature or attribute). Information entropy, H, is the amount of expected uncertainty found in a random variable, X, whose value we will refer to as the *class* of the data point. In D-TIPS, each data point is an action performed by a student and its class is the corresponding student group. IG when used in classifiers, such as decision trees [13], is applied to a dataset where each data point has multiple features in addition to its class. The IG of a given feature is then the reduction in expected uncertainty about the correct class of a data point when its feature value is known. IG is calculated as the difference between the information entropy of the data without knowledge of the feature values, $H(X)$, (i.e., based solely on the probability distribution of classes over the full dataset) minus the conditional entropy of the data set when the value of the feature is known, $H(X|F)$. In D-TIPS, the features are the patterns of actions to be ranked, and a particular action's feature value is the combination of whether the action begins an occurrence of the pattern, o, and the order number of the bin in which the action occurred, b. Therefore, the D-TIPS IG metric can be represented as:

$$IG(X|F) = H(X) - \sum_{v \in Vals(F)} p(F = v)H(X|F = v)$$

where

$$Vals(F) = \{(o, b)|o \in \{true, false\}, b \in \{1 \ldots n\}\}$$

Information gain is leveraged in classifiers to determine which features are most discriminatory because they provide the least amount of uncertainty among classes in the data. In a similar fashion, D-TIPS applies information gain to determine which patterns are the most interesting because knowledge of their occurrence *and* temporal location provides the least amount of uncertainty among the student groups. This information-theoretic definition of the D-TIPS measure provides two important properties: (1) given two patterns with the same total occurrences for each corresponding group (e.g., group A has an occurrence of a and group B has an occurrence of b for both patterns, although a and b may be different values), the pattern with the greater discrimination of groups by *differences in temporal location/bin among groups* will have a higher rank, and (2) given two patterns with the same relative temporal behaviors (i.e., the same proportion of a given group's total pattern occurrence in each corresponding bin) for each corresponding group, the pattern with the greater discrimination of groups by *differences in total occurrence among groups* will have a higher rank.

The D-TIPS measure provides a way of recognizing differences among groups both by total pattern occurrence and by temporal behavior (e.g., decreasing usage versus increasing usage, or spikes in different bins). Further, when the same differences across groups (by both total pattern occurrence and temporal behavior) occur for two patterns, the pattern with higher overall frequency will have the higher rank. Thus, D-TIPS tends to emphasize patterns with large relative differences among groups (by total occurrence and/or temporal behavior) even when they are not especially frequent in the overall dataset, while also emphasizing patterns with more moderate differences among groups when the frequency of the pattern in the overall dataset is high. Conversely, D-TIPS tends to deemphasize patterns that are homogeneous across groups (by both relative occurrence and temporal behavior) or that are especially rare in all groups.

3.4 Visualizing Temporal Evolution

Given a set of highly-ranked behavior patterns, researchers will need to analyze them in further depth to understand their implications for student learning and generate additional hypotheses or changes to scaffolding in a learning environment. An important aspect of this analysis is the consideration of how a pattern's frequency changes over time in different groups. In order to more rapidly and efficiently analyze this aspect of the patterns identified by D-TIPS, we employ a heat map visualization.

Heat maps utilize the counts of data (pattern occurrence frequency in this case) in discrete cells across one or more dimensions to determine the color of the cell. Cell color is based on where the corresponding count falls between the highest and lowest count in any of the cells. The nature of the coloring scheme helps draw a user's eye to areas of contrast and thus larger changes, as well as general trends, in the data. For this analysis of D-TIPS patterns, we employ a two-dimensional heat map where the x-axis is time/temporal-bin, while the y-axis is student group. Further, rather than raw occurrence counts, each cell's count is the *percentage* of its group's total pattern occurrence that falls within that temporal bin. The use of percentages of pattern occurrence allows analysis of temporal variation normalized by the total frequency of the pattern per group. Therefore, different temporal trends in pattern usage across groups will be highlighted, even when total pattern occurrence differs significantly among groups, which would otherwise wash out any trends in the groups with lower pattern occurrence.

4 Betty's Brain Data

The data employed for the analysis in Sect. 5 consists of student interaction traces from the Betty's Brain [21,22] learning environment. In Betty's Brain, students read about a science process and teach a virtual agent about it by building a causal map. They are supported in this process by a mentor agent, who provides feedback and support for their learning activities. The data analyzed here was obtained in a recent study with 68 7^{th}-grade students taught by the same teacher in a middle Tennessee school. At the beginning of the study, students were introduced to the science topic (global climate change) during regular classroom instruction, provided an overview of causal relations and concept maps, and given hands-on training with the system. For the next four 60-minute class periods, students taught their agent about climate change and received feedback on content and learning strategies from the mentor agent.

The study tested the effectiveness of two support modules designed to scaffold students' understanding of cognitive and metacognitive processes important for success in Betty's Brain. The *knowledge construction* (KC) support module scaffolded students' understanding of, and suggested strategies for, constructing knowledge by identifying causal relations in the resources. The *monitoring* (Mon) support module scaffolded students' understanding of, and suggested strategies for, monitoring Betty's progress by using the quiz results to identify correct and incorrect causal links on Betty's map. Participants were divided into a control and three treatment groups. The knowledge construction (KC) group used a version of Betty's Brain that included the KC support module and a causal link tutorial that they could access at any time and were prompted to enter when the mentor determined they were having difficulty identifying causal links in the resources. The monitoring (Mon) group used a version of Betty's Brain that included the Mon support module and a tutorial about employing link annotations to keep track of links shown to be correct by quizzes. The full (Full) group used a version of Betty's Brain that included both support modules and tutorials. Finally, the control (Con) group used a version that included neither the tutorials nor the support modules.

In Betty's Brain, the students' learning and teaching tasks were organized around seven activities: (1) reading resource pages to gain information, (2) adding or removing causal links in the map to organize and teach causal information to Betty, (3) querying Betty to determine her understanding of the domain based on the causal map, (4) having Betty take quizzes that are generated and graded by the mentor to assess her current understanding and the correctness of links in the map, (5) asking Betty for explanations of which links she used to answer questions on the quiz or queries, (6) taking notes for later reference, and (7) annotating links to keep track of their correctness determined by quizzes and reading. Actions were further distinguished by context details, which for this analysis were the correctness of a link being edited and whether an action involved the same subtopic of the domain as at least one of the previous two actions. The definition of actions in Betty's Brain learning activity sequences are discussed further in [6].

5 Results

To illustrate and characterize the performance of the D-TIPS technique, we present selected results from its application to student learning activity data in the Betty's Brain classroom study described in Sect. 4. The first step of the D-TIPS analysis identified 560 candidate patterns that occurred in at least half of the students in one or more of the four experimental conditions. Given the limited number of students in each condition, we chose to bin pattern occurrence values into fifths of the activity sequences for a broad analysis of their usage evolution over time. Table 1 presents 3 of the top 30 most differentially-interesting patterns identified by D-TIPS across the four scaffolding conditions. For comparison, the average occurrences per student and ranking by that value is also presented. Over half (18) of the top 30 D-TIPS patterns had a rank past 50th by occurrence, with 13 of them ranking beyond 100th, indicating that they would be unlikely to have been considered and further analyzed without D-TIPS.

Table 1. Selected patterns with D-TIPS and occurrence rankings

Pattern	D-TIPS rank	Occurrence rank	Avg occurrence
[Quiz]	3	2	21.8
[Read] → [Note]	18	100	1.7
[Read] → [Read] → [Remove Link$^-$]	29	137	1.4

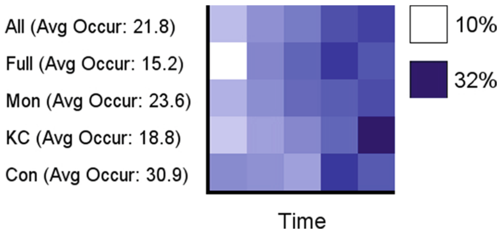


Fig. 1. [Quiz]

The first pattern in Table 1 illustrates a single action pattern that was ranked very high by both D-TIPS and overall occurrence. While individual student actions are often less interesting than longer patterns, they are still important to consider, especially when they also illustrate a tendency to be employed differentially across groups and over time. Figure 1 shows that all groups tended to use quizzes more frequently later in their work on the system. Since students' causal maps grew over time, monitoring and correction of the maps were more important later in their learning activities. There were some differences in usage trends over time among the different conditions, such as the steeper increasing trend for the KC and Full groups than the Monitoring group and the somewhat earlier peak in usage for the Full and Control groups. However, the overall occurrence by conditions differed markedly, with the Control group performing far more quiz actions than the others, and the Monitoring group performing more quiz actions than the KC and Full groups. While the Monitoring group's use of the quiz was expected to be high due to the focused monitoring support that relied heavily on the quiz, it is surprising that the Control group had the highest quiz usage. In addition, the Control group illustrated less of an increasing trend in usage compared to other groups, employing quizzes nearly as much in the first two fifths of their activity as in the final fifth. These results may indicate that without either KC or monitoring support, the Control group struggled more and fell back on strategies of guessing and checking (with the quiz).

Figure 2 illustrates a knowledge construction behavior of reading and taking notes that was ranked highly by D-TIPS. Another difference among the groups, which added to the interestingness of this pattern under the D-TIPS analysis, is that the Control group tended to perform reading followed by note-taking primarily in the last fifth of their activities, as opposed to the first two fifths for the other groups. Further analysis of the data attributed this primarily to two of the Control group students, although the reason for this aberration is still unclear.

The pattern illustrated in Fig. 3 involves a sequence of (two) reading actions followed by removing an incorrect link. While there was no consistent temporal trend in the usage of this pattern, the Monitoring and Control groups exhibited this pattern less than once per student, while the KC group averaged 2.4 times per student. Although ranked lower by D-TIPS at 45th, the sub-pattern of a

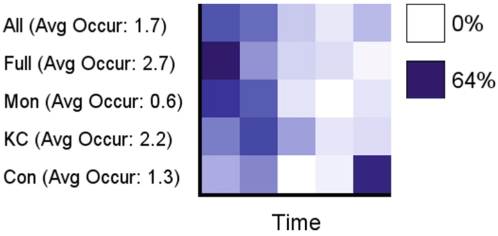


Fig. 2. [Read] → [Note]

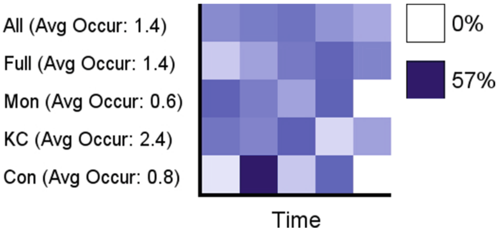


Fig. 3. [Read] → [Read] → [Remove Link$^-$]

single read action followed by removing an incorrect link illustrates the same differences. This suggests that students with the KC feedback generally relied more heavily on reading to identify incorrect links than either the Control and Monitoring groups, possibly because the Control group struggled more in general and the support in the Monitoring group focused students more on the use of quizzes to identify incorrect links.

6 Conclusion

While identification of high-frequency behavior patterns is undoubtedly useful, finding patterns that have differing occurrence over time across a set of student groups is also important for analyzing learning behaviors and the effects of scaffolding. In this paper, we presented the D-TIPS interestingness measure and mining approach, which identifies patterns that differ in their usage among student groups by either total (group) occurrence or temporal behavior, even when they are not especially frequent in the overall dataset. Results from the use of this technique to mine Betty's Brain data illustrated the potential benefits and helped characterize differences between D-TIPS and a baseline occurrence ranking. Moreover, D-TIPS identified patterns that illustrated potentially important differences in learning behavior among different scaffolding conditions that would have probably been overlooked by considering only pattern frequency. Future work will include autonomous identification of an effective number of bins for splitting a given set of activity sequences, as well as methods to individually characterize student groups by the patterns identified in D-TIPS. Further, we intend to apply the D-TIPS analysis to data in other domains to illustrate its generality and utility in areas beyond education.

Acknowledgments. This work has been supported by NSF-IIS Award #0904387 and IES Award #R305A120186.

References

1. Kinnebrew, J.S., Mack, D.L., Biswas, G.: Mining temporally-interesting learning behavior patterns. In: DMello, S.K., Calvo, R.A., Olney, A. (eds.): Proceedings of the 6th International Conference on Educational Data Mining, pp. 252–255 (2013)
2. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings of the Eleventh IEEE International Conference on Data Engineering (ICDE), pp. 3–14 (1995)
3. Zaki, M.: Sequence mining in categorical domains: incorporating constraints. In: Proceedings of the Ninth International Conference on Information and Knowledge Management, pp. 422–429. ACM (2000)
4. Nesbit, J., Zhou, M., Xu, Y., Winne, P.: Advancing log analysis of student interactions with cognitive tools. In: 12th Biennial Conference of the European Association for Research on Learning and Insruction (EARLI) (2007)
5. Perera, D., Kay, J., Koprinska, I., Yacef, K., Zaïane, O.: Clustering and sequential pattern mining of online collaborative learning data. IEEE Trans. Knowl. Data Eng. **21**(6), 759–772 (2009)
6. Kinnebrew, J.S., Loretz, K.M., Biswas, G.: A contextualized, differential sequence mining method to derive students' learning behavior patterns. J. Educ. Data Min. **5**(1), 190–219 (2013)
7. Kinnebrew, J.S., Biswas, G.: Identifying learning behaviors by contextualizing differential sequence mining with action features and performance evolution. In: Proceedings of the 5th International Conference on Educational Data Mining (EDM 2012), Chania, Greece, June 2012
8. Amershi, S., Conati, C.: Combining unsupervised and supervised classification to build user models for exploratory learning environments. J. Educ. Data Min. **1**(1), 18–71 (2009)
9. Martinez, R., Yacef, K., Kay, J., Al-Qaraghuli, A., Kharrufa, A.: Analysing frequent sequential patterns of collaborative learning activity around an interactive tabletop. In: Proceedings of the Fourth International Conference on Educational Data Mining, Eindhoven, Netherlands (2011)
10. Su, J.M., Tseng, S.S., Wang, W., Weng, J.F., Yang, J., Tsai, W.N.: Learning portfolio analysis and mining for scorm compliant environment. J. Educ. Technol. Soc. **9**(1), 262–275 (2006)
11. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: a survey. ACM Comput. Surv. (CSUR) **38**(3), 9 (2006)
12. Zhang, Y., Zhang, L., Nie, G., Shi, Y.: A survey of interestingness measures for association rules. In: International Conference on Business Intelligence and Financial Engineering, BIFE'09, pp. 460–463. IEEE (2009)
13. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2007)
14. Mitchell, T.: Machine Learning. McGraw Hill, New York (1997)

15. Merceron, A., Yacef, K.: Interestingness measures for association rules in educational data. In: Educational Data Mining 2008, p. 57 (2008)
16. Kotsiantis, S., Pierrakeas, C.: Efficiency of machine learning techniques in predicting students performance in distance learning systems. Technical report, Citeseer.
17. Sotiriou, C., Wirapati, P., Loi, S., Harris, A., Fox, S., Smeds, J., Nordgren, H., Farmer, P., Praz, V., Haibe-Kains, B., et al.: Gene expression profiling in breast cancer: understanding the molecular basis of histologic grade to improve prognosis. J. Natl Cancer Inst. **98**(4), 262–272 (2006)
18. Mashima, D., Kobourov, S., Hu, Y.: Visualizing dynamic data with maps. IEEE Trans. Vis. Comput. Graphics **18**(9), 1424–1437 (2012)
19. Ho, J., Lukov, L., Chawla, S.: Sequential pattern mining with constraints on large protein databases. In: Proceedings of the 12th International Conference on Management of Data (COMAD), pp. 89–100 (2005)
20. Renyi, A.: On measures of entropy and information. In: Fourth Berkeley Symposium on Mathematical Statistics and Probability, pp. 547–561 (1961)
21. Biswas, G., Leelawong, K., Schwartz, D., Vye, N., Vanderbilt, T.: Learning by teaching: a new agent paradigm for educational software. Appl. Artif. Intell. **19**(3), 363–392 (2005)
22. Leelawong, K., Biswas, G.: Designing learning by teaching agents: the Betty's Brain system. Int. J. Artif. Intell. Educ. **18**(3), 181–208 (2008)

Mining Top-K Relevant Stay Regions from Historical Trajectories

Yung-Hsiang Lin[1], Chien-Hsiang Lai[1], and Po-Ruey Lei[2](✉)

[1] Department of Computer Science, National Chiao Tung University,
Hsinchu, Taiwan, ROC
{shiang1095,likekaito}@gmail.com
[2] Department of Electrical Engineering,
ROC Naval Academy, Kaohsiung, Taiwan, ROC
cnabarry@gmail.com

Abstract. With increasingly prevalent mobile positioning devices, such as GPS loggers, smart phones, and GPS navigation devices, a huge amount of trajectories data is collected. Users are able to obtain the various location-based services by uploading their trajectories. In this paper, we address that a user's movement behavior is able to discover by their similar shape trajectories and resulted in some regions frequently stay in common, called relevant stay regions. Once a set of stay regions discovered, we can predict the next region where the user intends to go and provide location-based information of the next stay in advance, such as traffic status, targeted advertises, sightseeing recommendations, and so on. Prior works have elaborated on discovering stay region from the whole crowd trajectories and then exploring the relations between the regions to describe the movement patterns for location prediction. However, the trajectories pass the same region may not have the similar movement behavior. Thus, we propose a framework to discover stay regions relevant to the specific movement behavior and then applied in location prediction, called Region Modeling and Mobility Prediction. The proposed framework includes two modules: region modeling and mobility prediction. In the region modeling module, we develop shape clustering method to group the similar trajectories from historical data and then explore the stay region model from trajectory clusters. Based on the discovered region model, the mobility prediction module provide a cluster selection algorithm and several prediction strategies to generate the top-k relevant stay regions. Experiments results on real datasets demonstrate the effectiveness and accuracy of our proposed model on detecting next stay region, comparing with other baseline methods.

1 Introduction

While the mobile positioning devices become prevalent, a tremendous amount of trajectory data is generated. Users can upload their trajectories or check-in data to location-aware web provider(e.g., @trip, Foursquare and Facebook) and obtain the various location-based services such as tourism recommendations and

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 293–304, 2014.
DOI: 10.1007/978-3-319-13186-3_28

store advertising in their daily life. In general, trajectory data is a sequence of GPS points and these sequential GPS points record the users' true movement. If some trajectories frequently and repeatedly appear in a user's historical trajectory data, we are able to suppose that the user may have a certain movement behavior. However, due to the uncertainty of GPS position collected, it is difficult to discover that the trajectories are completely repeated. For example, in Fig. 1, there are three historical trajectories of one user in different days with different colors. These trajectories are not completely the same but pass through some regions in common. More specifically, those trajectories have similar shape and all of them have stayed in some specific regions together(Region R1, R2, and R3). This observation shows that the user usually takes similar routes if they have visited the similar set of stay regions. Thus, we claim that a user's movement behavior can be discovered by clustering the historical trajectories with similar shape and then resulted in some stay regions where the user frequently visited. The sets of stay regions which discovered from the similar trajectories are relevant to a user's movement behavior and called relevant stay regions. Once a set of stay regions discovered, lots of location-aware information and applications could be provided to user, such as inferring regions for sightseeing and tourism recommendations, distributing coupons of stores near by stay location, estimating the traffic status on the way to destination, and even predicting the possible next stay for navigation system to set the destination automatically. In this paper, we focus on the problem of mining stay regions from historical trajectories and applied in location prediction problem.

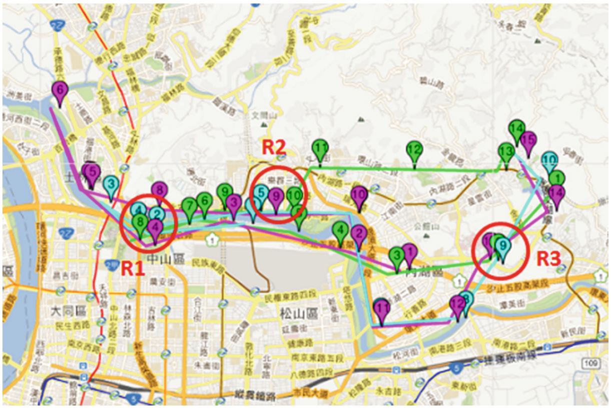


Fig. 1. An example of the similar-shape historical trajectories and passed some stay regions in common.

Given a set of trajectories, prior works have studied the location prediction problem in which given the users current location, the problem is to predict the next location or the location at a specific time. In [7,10], the authors first map the user's historical trajectories into the regions by grid-cell system and estimate the transition probability between the mapped regions. Then, according to the user's current location to predict the future movement based on the probability model. The authors in [5,9] construct a decision tree according to trajectories and use tree structure to denote the stay locations for prediction. These existing research works focus on discovering stay region from the whole crowd trajectories and then exploring the relations between the regions to describe the movement patterns for location prediction. However, the trajectories stay in the same region may not have the similar movement behavior. For instance, a user usually stays a restaurant for lunch in the daytime and then go back to work. In the nighttime, the user sometimes exercises at the gym in the same region and then go back home. Both trajectories stay in the same region but those are two different movement behaviors. In this paper, we first discover the user's movement behavior by trajectory clustering and then explore the stay regions from each group of trajectories. Based on those sets of stay regions from the similar movement behavior, the next stay region prediction is able to be improved. The advantage of predicting the future location by the stay regions generated from the similar movement behavior is that the regions where the user frequently stay are the same even if the user detours his/her route for traffic or other instance.

In this paper, we propose a framework to discover stay regions relevant to the specific movement behavior and then applied in location prediction, called Region Modeling and Mobility Prediction. Specifically, the framework includes two modules: region modeling module and mobility prediction module. In region modeling module, given a use's individual trajectory data, we develop the shape clustering method to discover the user's movement behavior by grouping the trajectories with the similar movement shape. Based on the trajectory clusters, a stay regions model is discovered for each trajectory cluster. In mobility prediction, given the starting location and current time, top-k stay regions relevant to the user's movement behavior will be provided. We design an algorithm to select trajectory cluster, i.e. the specific movement behavior, by considering the user's starting location and current time. Then, several prediction strategies are proposed to generate top-k stay regions those relevant to the user's specific movement behavior. We evaluate the performance of our proposed framework by real-world dataset generated by mobile users in Taiwan. In addition, we also compare with the other existing approaches. The extensive experiments results demonstrate the effectiveness of our framework and the accuracy of next stay region prediction ia able to be improved.

The contributions of this paper are summarized as follows:

1. We address that a user's movement behavior can be discovered by clustering the historical trajectories with similar shape and then resulted in some regions where the user frequently stayed. Such a set of stay regions is relevant to a specific movement behavior, called relevant stay region.
2. We propose a framework, called Region Modeling and Mobility Prediction, to discover movement behavior by proposed trajectory shape clustering and predict top-k relevant stay regions where a user intends to go.

3. Extensive experiments are conducted on real datasets to evaluate our proposed framework. The results demonstrate that our framework is more effective and accuracy than existing works.

The rest of this paper is organized as follow: Sect. 2 introduces the current research works of the location and destination prediction. Section 3 states the problem and gives an overview of our proposed framework. Section 4 presents the shape clustering and region modeling on historical trajectory. Section 5 describes mobility prediction module for top-k stay regions generation. Performance studies are presented in Sect. 6. Finally, Sect. 7 concludes this paper.

2 Related Work

There are many research works discussing the problem of location prediction, they only focus on discovering the frequent region from the whole crowd trajectories and then exploring the relations between the regions to describe the movement patterns for location prediction. However, the trajectories stay the same region may not have the similar movement behavior. A stay region is supposed to be relevant to a movement behavior, in other words, the stay regions should be generated from similar trajectories. The route to next stay region may be detoured due to traffic jams and other instances. Furthermore, the next region where the user frequently move to may not be changed according to the user's movement behavior. Therefore, those relevant stay regions can be adopted to improve the accuracy of location prediction. In this section, we first discuss some research works on the location prediction and then introduce the research works related to mining stay regions.

A number of location prediction techniques have been proposed in data mining literature. Markov model has been widely applied in predicting destinations for a specific individual as well [1,10]. In [10], the author uses a Markov model to offline prepare the probabilities needed to efficiently compute the posterior probability for any given query trajectory online. Some existing works use the external information to predict destination [6,12], these external information such as the distributions of different districts (ground cover), of traveling time, of trajectories length, the accident reports, road condition, and driving habits often enhance the prediction accuracy. Even context information such as time-of-day, day-of-week, and velocity has been incorporated as the features in training the Bayesian network model for prediction [4]. Chen et al. [2] used a tree structure to represent the historical movement patterns and then matched the current partial trajectory by stepping down the tree. Trajectory pattern [5,9] are first explored and apply on location prediction. There are some existing works focus on extracting the stay regions for semantics mining. The authors in [11] proposed the concept of stay point detection to discover the stay regions. Considering the both spatial and temporal information, the authors in [8] propose a sequential clustering method to extract the stay regions.

3 Framework Overview

A user's movement behavior can be discovered by clustering the historical trajectories and then resulted in some regions where the user frequently stayed. Such a set of stay regions relevant to the movement behavior, called *relevant stay region*. We claim that the users' stay regions will be relevant to their movement behavior. Based on the discovered sets of relevant stay regions, the prediction on next stay region can be improved. Thus, we propose a framework to discover movement behavior and generate relevant stay regions and then applied in location prediction, called Region Modeling and Mobility Prediction.

Figure 2 shows the proposed framework which is comprised of two components: Region Modeling Module and Mobility Prediction Module. Since there are two subtasks in the region modeling module, exploring movement behavior and detecting the relevant stay regions. We first propose a trajectory clustering method called shape-clustering to explore the movement behavior by grouping the similar-shape trajectories. Based on the trajectory clusters, the sets of relevant stay regions are discovered. Specifically, a set of stay regions which is relevant to specific movement behavior is discovered from each trajectory cluster. For detecting the relevant stay regions of each trajectory cluster, we detect stay points from each trajectory and then adopt a Share Nearest Neighbor clustering(SNN-clustering) to cluster the stay points as stay regions. Finally, this module generates a relevant region prediction model for next stay region prediction. In mobility prediction module, we design an algorithm to select trajectory cluster by considering the user's starting location and current time, i.e. determining the specific movement behavior by the user's current movement. Then, several prediction strategies are proposed to generate top-k stay regions those relevant to the user's specific movement behavior.

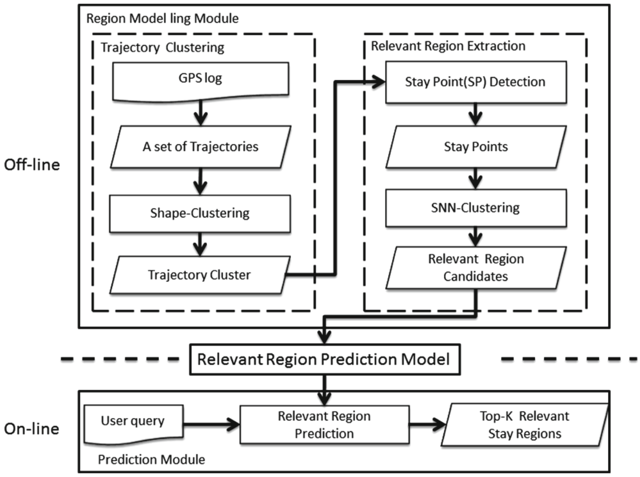


Fig. 2. Framework

4 Movement Behavior Discovery and Region Modeling

In this section, we will discuss how to discover the relevant region model for next region prediction. This includes two subproblems, namely, discovering movement behavior from trajectories and mining relevant stay region by these behaviors. To achieve the goal, we first propose Shape-Clustering to explore the group of the similar-shape trajectories. Then, based on the discovered trajectory clusters, we develop a method to extract the relevant stay region from these trajectory clusters.

4.1 Shape-Clustering

The first step in our framework is to find the movement behavior by clustering the trajectories with the similar movement. More specifically, we attempt to cluster the similar-shape trajectories so as to be able to represent the user's movement behavior. In existing work, a shape-based pattern detection method has been used to detect streaming time series data [3]. We adopt the concept on trajectory data to find out the similar-shape trajectory. The similar-shape trajectories imply that these trajectories have the similar movement behavior. As shown in Fig. 3, there are two trajectories T_i and T_j denoted as sequential GPS points. Then, we define $segment_{i,n}$ as a nth trajectory segment in T_i in the defined time interval. To avoid the problem of trajectory segmentation [8], each segment has a overlap to smooth the segmentation. We calculate the average position(the two red points) in each segment according to latitude and longitude of GPS points in a segment. The distance d is calculated and defined as distance measure between the two average position(the two red points).

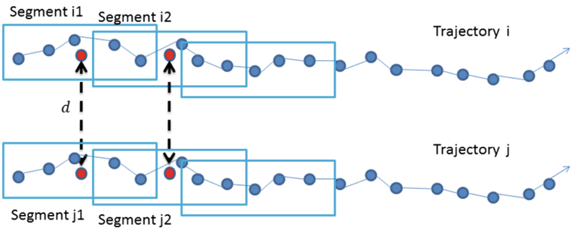


Fig. 3. An example of shape clustering

The shape clustering algorithm is shown in Fig. 4. If the distance d less than the distance threshold δ, these two segments are considered as similar segments. In addition, we define a similar counter mentioned as $SimSeg$ in the algorithm to count number of similar segments. Thus, the value of similar counter can be considered as similarity between two trajectories. Then, each two trajectories whose similarity is higher than similarity threshold $SimThres$ should be group into the same cluster. Otherwise, the trajectory form a new cluster independently.

Algorithm 1 Shape-Clustering Algorithm

Input: A trajectory $T_i = p_1 p_2 p_3 ... p_j ... p_{len_i}$, A trajectory $T_j = p_1 p_2 p_3 ... p_j ... p_{len_j}$, similarity threshold $SimThres$, distance threshold δ

Output: Trajectory clusters

```
$SimSeg$ = 0
$TotalSeg$ = 0
$TS_i \leftarrow$ timestamp of $p_1$ in $T_i$
$TS_j \leftarrow$ timestamp of $p_1$ in $T_j$
while $TS_i \leq$ timestamp of $p_{len_i}$ and $TS_j \leq$ timestamp of $p_{len_j}$ do
  $Avgpoint_i \leftarrow$ average of points in segments$[TS_i\ , TS_i + interval]$
  $Avgpoint_j \leftarrow$ average of points in segments$[TS_i\ , TS_i + interval]$
  $TotalSeg \leftarrow TotalSeg + 1$
  if distance between $Avgpoint_i$ and $Avgpoint_j \leq \delta$ then
    $SimSeg \leftarrow SimSeg + 1$
  end if
  $TS_i \leftarrow$ SegmentSmoothing($TS_i$)
  $TS_j \leftarrow$ SegmentSmoothing($TS_i$)
end while
if $SimSeg$ / $TotalSeg \geq SimThres$ then
  put $T_i$ and $T_j$ into same trajectory cluster
end if
```

Fig. 4. Shape clustering algorithm

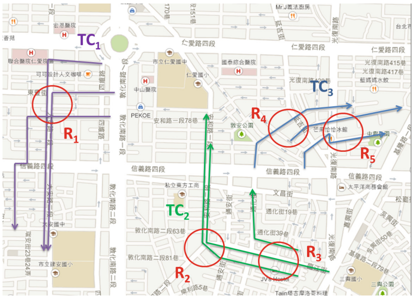


Fig. 5. An example of region modeling

4.2 Relevant Region Extraction

According to the result of shape-clustering, the second step is to detect possible relevant stay regions for each trajectory cluster. We adopt the conventional approach to extract stay region from trajectories [8,11]. We first detect the stay points from each trajectories. A stay point is detected when the consecutive points of a examined point do not exceed the predefined distance threshold during the specified period of time threshold. Then, the clustering method is applied to group the stay points those are close enough. The cluster of stay points is able to represent a region where the user frequently stays. A stay region is a summary of a set of similar stay points from different trajectories. To define the similarity between stay points and discover the stay regions, we adopt SNN (shared nearest neighbor)density-based clustering. When applying SNN density based clustering to discover stay regions, we constrain the searching range of nearest neighbors is a radius D_h around the examined nodes. We define a stay point is in a stay region if each stay point of which contains at least $MinSR$ number of neighbors in the distance radius D_h. The points without $MinSR$ nearest neighbors are viewed as non-stay points and discarded. All the connected components in the resulting graph are clusters finally. These clusters can be considered as stay region candidates where an object often stay for certain activities. After generating the relevant stay region candidates, to avoid the region formed by traffic jam, we check whether the region is on the road exactly or not. If stay regions are located on road, the stay regions are removed from candidates since the regions may be formed by traffic jam. For an example in Fig. 5, the result of this module are three trajectory clusters(TC_1, TC_2 and TC_3) and the trajectories in the same cluster have similar shape with each other. In each cluster, there are some relevant stay region candidates($R_1, R_2, ..., R_5$) generated by stay points detection and SNN-clustering.

5 Mobility Prediction

After extracting the relevant stay regions from trajectory clusters, a relevant stay region model for mobility prediction is constructed. Given a user's current location and time, the task of mobility prediction is to select the best trajectory cluster and determines the score of relevant stay region candidates respectively. Thus, the two-stage prediction module is proposed, including trajectory cluster selection and prediction strategy. Finally, according to the user's current location and time, the top-k relevant stay regions will be generated for next stay region prediction.

Due to a trajectory cluster is summarized from the similar-shape trajectories, we address that the trajectory cluster is able to imply a user's movement behavior. The next stay region prediction can be improved by estimate the user's next stay region on his/her movement behavior. By considering a user's current location and time, we propose a method to select the best trajectory cluster for prediction. Generally, movement behavior is usually relevant to time or location of a stay region. Thus, the average time and location of relevant stay region candidates in the trajectory cluster are determined by the score of the trajectory cluster. The average time and location of relevant stay region candidates are defined as Eqs. 1 and 2. We formulate the scoring function of a trajectory cluster as shown in the Eq. 3. The scoring value of the trajectory cluster which is closer to zero shows that the trajectory cluster has the better matching according to the user's current behavior. The trajectory cluster is selected as the candidate for generating the relevant stay regions.

$$TC_i.time = \frac{\sum_{j=1}^{n} SR_j.time}{n}, \forall SR_j \ in \ Traj \in TC_i \tag{1}$$

$$TC_i.loc = \frac{\sum_{j=1}^{n} SR_j.loc}{n}, \forall SR_j \ in \ Traj \in TC_i \tag{2}$$

$$score_{TC_i} = \frac{\frac{|TC_i.time - q.time|}{MAX(|TC.time - q.time|)} + \frac{d(TC_i.loc, q.loc)}{MAX(d(TC.loc, q.loc))}}{2} \tag{3}$$

Based on selected trajectory cluster, we evaluate each relevant stay region candidate which belong to the selected cluster and generate the top-k relevant stay region by their evaluated score. For scoring the potential of a relevant stay region where the user may move to, we design several prediction strategies by considering the user's current location and time. Three prediction strategies are proposed: (1) *Near Time First*, (2) *Near Location First* and (3) *High Frequency First*. *Near Time First*(Eq. 4) means that if the historical stay time on the stay region is closer to current time, the score is higher.

$$score_{NT,i} = \frac{1}{log_2(\Delta Time + 2)} \tag{4}$$

In the same manner, *Near Location First*(Eq. 5) denotes that the distance between stay region and current location, the score is higher.

$$score_{NL,i} = \frac{1}{log_2(\Delta Distance + 2)} \tag{5}$$

The last one is *High Frequency First*(Eq. 6), the more times the user has been stay region, the score is also higher.

$$score_{Freq,i} = \frac{Frequency_i}{Frequency_{max}} \tag{6}$$

Moreover, we consider that different users may have good effect by using different score functions, so the weighted average of these three score functions will be adopted. As the Eq. 7, the combination of score functions can be set different α,β and γ for different user. In general, the next stay location often near the current location and the time at next stay location often near the current time, so α and β can be set larger than γ. Otherwise, if the user often stay some specific location many times so that we can use *High Frequency First* to predict the next stay region very well, then the γ can be set larger. After this section, we evaluate the effect of these three criterions and the combination of score functions in experiment section.

$$score_{Comb,i} = \frac{\alpha \times score_{NT,i} + \beta \times score_{NL,i} + \gamma \times score_{Freq,i}}{\alpha + \beta + \gamma} \tag{7}$$

6 Experimental Results

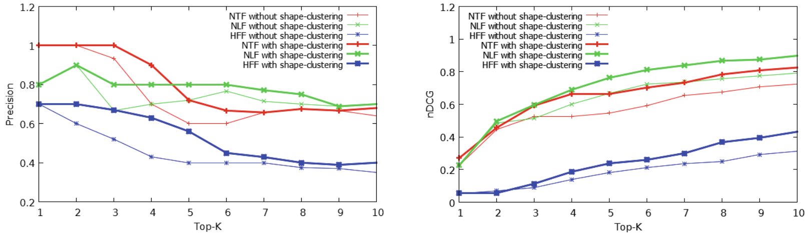
6.1 Dataset Description

For this study, we use a real-world trajectory datasets: trajectory data of trips from a website called @trip (http://www.a-trip.com/). @trip is a platform let users can upload their travel logs or check-in data and share these data for other users. We extracted a trajectory dataset which consists of 1,243 users, 14,039 trajectories and 13,192,283 GPS points. In order to test the effectiveness under various scenarios, the experiments are conducted on 10 selected users with different movement behaviors. Their stay regions relevant to different movement behavior are labeled as the groundtruth for evaluation.

6.2 Performance Evaluation

In this section, we first the evaluate the effectiveness of shape clustering on movement behavior discovery. Then, in order to show the improvement on prediction accuracy, we conduct the experiments compared with other existing methods.

Effectiveness of Shape Clustering. The shape-clustering method play a important role in our framework. We suppose that a user usually takes similar routes if they have visited the similar set of stay regions. Thus, a user's movement behavior can be discovered by clustering the historical trajectories with similar shape and then resulted in some stay regions where the user frequently visited. Such a set of relevant stay regions is able to apply on improving location prediction problem. In order to show the improvement on prediction accuracy, we compare the proposed framework using shape-clustering and without



(a) Comparison of precision

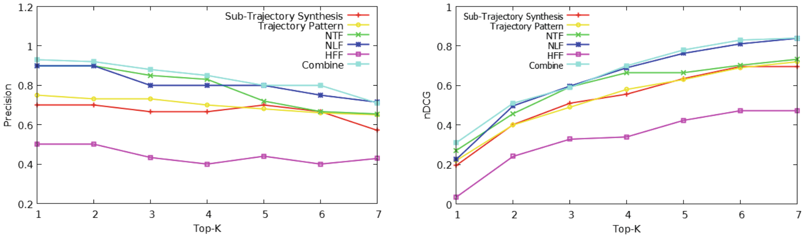
(b) Comparison of nDCG

Fig. 6. Effectiveness of shape clustering

using shape-clustering by evaluating their location prediction and top-k relevant region ranking. Figure 6(a) shows the precision of next stay region prediction. The precision of location prediction is improved. In Fig. 6(b), the accuracy of top-k relevant region ranking is evaluated by nDCG. The thick line denotes the framework using shape-clustering and thin line denotes the framework without using shape-clustering. The sorting result of proposed framework with shape-clustering has higher accuracy. We can observe the precision and nDCG are improved by using shape-clustering obviously. The experimental results shows the shape clustering is able to discover the movement behavior effectively and generate the top-k relevant stay regions precisely.

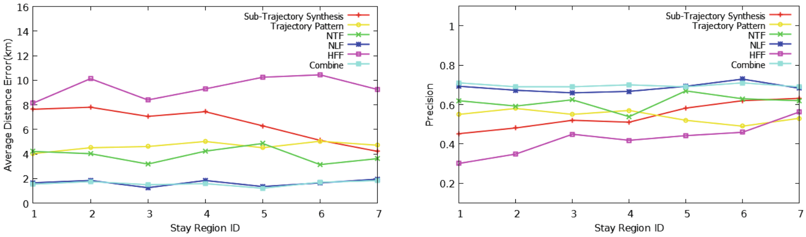
Evaluation of Next Region Prediction. We next test the accuracy of the next region prediction. Given a user's current location and time, the next region prediction is to forecast the next region where the user may possibly stay. We evaluate the prediction accuracy by the proposed framework with various scoring functions, called High Frequency First(HFF), Near Location First(NLF), Near Time First(NTF), and Combination(Combine). Additionally, we compare the performance with the existing methods [9,10]. In [10], the authors predict the destination by computing the posterior probability for any given query sub-trajectory(Sub-Trajectory Synthesis). In [9], they explore the trajectory patterns and construct the decision tree to predict the next stay location(Trajectory Pattern). To compare with them, we use average distance error and precision as measurements for the performance of next region prediction. In Fig. 7(a) and (b), although HFF can not get better effect, NLF and Combine methods have better precision and nDCG value than comparison targets.

Figure 8(a) shows the prediction results represented by average distance error. The experimental result shows that both Combine method and NLF have lower distance error than Sub-Trajectory Synthesis method and prediction by using Trajectory Pattern. We also evaluate the prediction of next location by precision. In Fig. 8(b), we can figure out the propose Combine method has higher precision even the predicting region is getting distance. Furthermore, in Fig. 8(b), the proposed methods NLF and Combine have higher precision than others.



(a) Comparison of precision (b) Comparison of nDCG

Fig. 7. Prediction accuracy comparison



(a) Average Distance Error (b) Precision

Fig. 8. Average distance error and precision for next stay region prediction

7 Conclusion

In this paper, we address that a user's movement behavior can be discovered by clustering the historical trajectories with similar shape and then resulted in relevant stay regions where the user frequently visited. We propose a framework, Region Modeling and Mobility Prediction, to generate top-k relevant stay regions from movement behavior and apply on location prediction problem. In region modeling module, the proposed shape clustering method explores the user's movement behavior by grouping the trajectories with the similar movement shape. Based on the trajectory clusters, a stay regions model is discovered for each trajectory cluster. In mobility prediction, given the starting location and current time, top-k stay regions relevant to the user's movement behavior will be provided for predicting next region. Experiments based on real datasets have shown that proposed framework is able to explore the movement behavior effectively and generate top-k relevant stay regions accurately. Furthermore, the ability to predict next region has advanced than the prediction methods in the literature.

Acknowledgment. Po-Ruey Lei was supported in part by the National Science Council, Project No. 102-2221-E-012-002.

References

1. Alvarez-Garcia, J., Ortega, J., Gonzalez-Abril, L., Velasco, F.: Trip destination prediction based on past gps log using a hidden markov model. Expert Syst. Appl. **37**(12), 8166–8171 (2010)
2. Chen, L., Lv, M., Chen, G.: A system for destination and future route prediction based on trajectory mining. Pervasive Mob. Comput. **6**(6), 657–676 (2010)
3. Chen, Y., Nascimento, M.A., Ooi, B.C., Tung, A.K.: Spade: on shape-based pattern detection in streaming time series. In: Proceedings of the IEEE 23rd International Conference on Data Engineering, pp. 786–795 (2007)
4. Gogate, V., Dechter, R., Bidyuk, B., Rindt, C., Marca, J.: Modeling transportation routines using hybrid dynamic mixed networks, pp. 116–131 (2005)
5. Jeung, H., Liu, Q., Shen, H.T., Zhou, X.: A hybrid prediction model for moving objects. In: Proceedings of the IEEE 24th International Conference on Data Engineering, pp. 70–79 (2008)
6. Krumm, J., Horvitz, E.: Predestination: Where do you want to go today? Computer **40**(4), 105–107 (2007)
7. Lei, P.R., Shen, T.J., Peng, W.C., Su, I.J.: Exploring spatial-temporal trajectory model for location prediction. In: Proceedings of the 12th IEEE International Conference on Mobile Data Management, pp. 58–67 (2011)
8. Lu, C.-T., Lei, P.-R., Peng, W.-C., Su, I.-J.: A framework of mining semantic regions from trajectories. In: Yu, J.X., Kim, M.H., Unland, R. (eds.) DASFAA 2011, Part I. LNCS, vol. 6587, pp. 193–207. Springer, Heidelberg (2011)
9. Monreale, A., Pinelli, F., Trasarti, R., Giannotti, F.: Wherenext: a location predictor on trajectory pattern mining. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 637–646 (2009)
10. Xue, A.Y., Zhang, R., Zheng, Y., Xie, X., Huang, J., Xu, Z.: Destination prediction by sub-trajectory synthesis and privacy protection against such prediction. In; Proceedings of the 29th IEEE International Conference on Data Engineering, pp. 254–265 (2013)
11. Zheng, V.W., Zheng, Y., Xie, X., Yang, Q.: Collaborative location and activity recommendations with gps history data. In: Proceedings of the 19th International Conference on World Wide Web, pp. 1029–1038 (2010)
12. Ziebart, B.D., Maas, A.L., Dey, A.K., Bagnell, J.A.: Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior. In: Proceedings of the 10th International Conference on Ubiquitous Computing, pp. 322–331 (2008)

Associative Classification for Human Activity Inference on Smart Phones

Yu-Hsiang Peng(✉), Gunarto Sindoro Njoo(✉),
Shou-Chun Li, and Wen-Chih Peng(✉)

Department of Computer Science, National Chiao Tung University,
1001 Ta Hsueh Road, Hsinchu 30010, Taiwan, Republic of China
{eric45670,gunarto.nctu}@gmail.com, wcpeng@cs.nctu.edu.tw

Abstract. With the population of smart phones, the general trend of human activity inference is prospering under a powerful computation capabilities on modern phones. Such an assistant make users life more convenient and help them prevent from unnecessary interferences. In conventional research, the activity inference problem is considered a classification instance, so in this paper we propose an association-based classifier framework (ACF) that aims at exploring the correlation among collected sensor data. Each data consists of multiple sensor readings with a label, e.g., dining, shopping, working, driving, sporting, and entertaining. Note that ACF caters to the discrete data; as a consequence, the continuous sensor readings are needed to be transformed to some discrete groups. Therefore, we propose an Interval Length-Gini Discretization (LGD) method which considers the groups and misclassified cases to obtain the best hypothesis for a given set of data. After an appropriate discretization, we propose one-cut and memory-iteration-based approach to select a set of useful sensor-value pairs for reducing the model size by removing redundant features and guaranteeing an acceptable accuracy. In the experiments our framework has a good performance on real data set collected from 50 participants in eight months, and a smaller size than the existing classifications.

Keywords: Activity recognition · Smart phones · Classification · Associative rule · Discretization · Feature selection

1 Introduction

Human activity recognition is an important issue. Many stream data mining methods have been proposed. Some research in [10] can accurately recognize activities if sensor data is collected from smart environments. Another research in [11] even can predict activity based on video stream data. Other researchers in [2,12] focus on how to recognize simple activities based on video data. But all these approach are not suitable to recognize user activities on smart phones since they need put sensors in stable environments but the sensors on smart phones often face different environments. In recent years, activity recognition

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 305–317, 2014.
DOI: 10.1007/978-3-319-13186-3_29

using mobile device is becoming more popular because smart phones has already equipped with various types of sensors, such as: several motion sensors (e.g. accelerometer sensors), environment sensors (e.g. light sensors, and pressure sensors), temporal sensors (e.g. time) and location sensors (e.g. GPS sensors). Previously, many researchers try to use these equipments to recognize user activities. The authors in [13] try to use wearable motion sensors attached in several human body position to recognize motion activities. Other authors in [1,6] use motion sensors on smart phones to detect some simple motion activities, such as walking, running, going up, and going down. Some of them even can roughly detect complex motion activities. These work can achieve high accuracy in mobile device but all of them do not consider the storage issue in smart phones. So, we want to design a framework which can detect user complex life activities and it is suitable on smart phones.

Human activity has lots of types. In this paper we focus on life activities (e.g. shopping, entertainment, sporting, working, transporting and dining). Comparing to action activities (e.g. walking, running, going up and etc.), because life activities can be regarded as the combination of some action activities, life activities are more meaningful for human and detecting them is more challenging. The problem of activity inference can be deemed as classification problem. Logger collect training dataset to build the classifiers. Classifier use the sensor data to recognize user activities. To use classifier on smart phones, there are some limitations. First, recognizing time need to be as short as possible in order to enhance user experience. Then, model size need to be as small as possible so that the model can be built in low layer architecture (e.g. sensor hub [7]). Associative classifier conforms the above two requirements. The detail of associative classifier will be introduced in Sect. 3.1. Associative classifier aims at mining association rules among context information hidden in the training dataset. In general, the classifier composed of association rules has smaller model size than other model. In addition, rule-based classification has another advantage. By using rule-based classification, users can understand their behavior easily by looking at rules in the model. Hence users can observe the model to realize their lifestyle and behavior.

Some sensor data from mobile device are continuous value. So, another issue is how to discretize these data. Discretziation method will influence the performance and model size of classifier. Discretization method has already been a mature research topic. There are two issues in discretization which are how to judge the quality of partition and what is stop condition. For first issue, in this paper, we use two famous discretization method idea. One is Gini index which is proposed by C. W. Gini [5]. Another one is entropy-based discretization standard (e.g. information gain). After these standard had been published, one research compares the performance of these methods [9]. Experiment shows that it is hard to determine which one is better. For second issue, in our observation, having too many slots will cause the classification algorithms to be cumbersome and having too few slots will cause the redundancy and impurity of slots. Therefore, we decide to propose a method called Length-Gini Discretization (LGD) which uses gini to test the quality of slot and also consider number of intervals.

Furthermore, the data which are gathered from sensors are also not always relevant and important to activity recognition learning algorithm because some sensor-recording values are weakly related to activity detection. In real world situation, using all features may produces adverse effect on training process, because of redundant or noisy features. Because of it, feature selection is needed to improve the quality of the data. Furthermore, feature selection is also able to reduce the model size and it is very important as that the resource and space in the mobile device is limited. We see each sensor-value pair, which is the outcome after discrete step, as a feature. In this paper, we use entropy-based feature selection [14] approach to select the feature rather than solely choosing it based on its coverage only. The selection of the feature is also involving learning algorithm such as Naive Bayesian which is used to evaluate selected features. One approach is using one-cut entropy threshold as the selector of the features and cut every features which not satisfy the threshold. We also propose another approach that is greedy-iteration technique which is able to select the features using several iterations. Because of it, greedy approach is able to preserve some feature which is being eliminated in the one cut approach.

Our main contributions in this paper are:

- We propose a framework to recognize human activity behavior by using smart phone sensors. We focus more on life activities instead of motion activities, as it is more meaningful for human.
- We designed a LGD approach to discretize the feature data by using Gini index and number of intervals. This approach has good compatibility with the rule-based classifier and it can improve the accuracy of classifier.
- We designed an iteration-based feature selection approach with the help of entropy as a measure which is able to select several features that have special characteristics, to overcome the limitation of using one-cut approach.

2 A Framework of Activity Inference

In this paper, we use android phone as a platform to detect human activity. Our collected data is a sequence of sensor data d in the form $d = < t, S >$ where t denotes a timestamp, and S denotes a sensor ID and its values. Our output in mobile device is the activity of user. Figure 1 shows the overall architecture of our activity recognition system. As figure shown, this work is consist two parts, off-line part and on-line part.

In the off-line phase, we do the training work. There are four main tasks in this phase called Feature Extraction, Data Discretization, Feature Selection and Model Construction. In Feature Extraction, every sensor has different sampling rate, so each record data d do not necessarily always have complete sensor data. We fill the missing sensor data by adding *Non-Data*. After that, we extract four types of feature called motion feature, location feature, temporal feature and

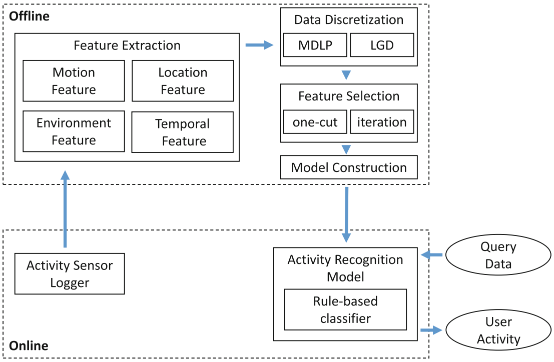


Fig. 1. Proposed framework of activity recognition in smart phones.

environment feature. Each feature contributes different meaning in varied activities and users. After extracting features, we need to discretize those features as our activity recognition algorithm only processes categorical data. In this paper, we adopt one method called MDLP as comparison and propose one method called LGD. MDLP uses the idea of minimum description length to cut the interval and let each interval has significant meaning. LGD consider the data distribution and feature's cooperation. Each method has its own advantages. The detail of these two methods will be introduced in Sect. 4. Another consideration is that some features are meaningless to some user. Thereby we also do feature selection to let the features personalized and even more it is also able to reduce the model size. In feature selection phase, we use one-cut approach and memory-iteration-based approach to select the features. In the last step, we construct rule-based classifier. Rule-based classifier has some advantages which make it suitable for recognizing activity in mobile device. The detail of the model and its advantages will be introduced in Sect. 3.

In the on-line phase, we do the recognizing work. Mobile device uses the result from feature selection phase to determine which feature should be collected and sent to the classifier. The classifier use the model, which is built in the off-line phase, to recognize the user activity and show in the application. For example, Table 1 show how the model is in mobile device. If the sensors collect that the value of GPSspeed is 5.5 m/s and the value of AccelSd is 2.7 m/s^2, the application in mobile device will recognize that the user's current activity is transportation.

Table 1. Example of rule-based classifier model

Priority	Rule
1st	GPSspeed = (0–6] and AccelSd = (0–4] − > Transportation
2nd	GPSx = (0.1–0.4] and GPSy = (0.0–0.2] − > Working
3rd	AccelAvg = (6–8] and AccelSd = (2–4] and GyroSd = (2–4] − > Sporting

3 Activity Recognition

In this section, we introduce what kind of model is suitable for smart phones by analyzing its characteristic. Then we introduce the challenge of using this model and how we conquer it. Finally, we introduce how to build model and set the parameter.

3.1 Recognition Model for Smart Phones

There are lots of ways to build recognition model. Some of them belong to machine learning, like SVM and Neural Networks. Some of them belong to traditional classification, like bayesian probability model, decision tree and k-Nearest Neighbor (k-NN). Each of them has its own advantages and disadvantages. Determining a good classifier in mobile device itself is a challenge.

There are some characteristics that are required to use classification model in the smart phones. First, the model size need to be as small as possible so that the recognition model can be built in low layer architecture such as *sensor hub*, which is a microcontroller unit that help integrate data from different sensors and process them [7]. Since some sensor hub has only 16 KB RAM so model size is an important issue. Another issue is that in order to enhance user experience to its maximum, then the model need to recognize the activity as fast as possible. In this paper, we find that associative classifier satisfies both request above. The recognition model we used is Classification Based on Associations (CBA) [8]. It is composed by ordered associate rules which is shown in Table 1. We use this model by considering several reasons which is explained below.

- Rule-based uses straightforward approach to recognize the activity in on-line step. CBA uses only rule matching to classify so that it can reduce the time and resource for activity recognition.
- In most case, model size is smaller than bayesian probability model, decision tree and SVM, so it is suitable to be put in smart phone.
- Rule-based model is user friendly, so user can see the model to realize their behavior and modify it.

3.2 Model Building and On-line Recognition

Associative classifier is suitable for mobile device but there is still a challenge about how to use sensor data to build CBA model. In this paper, we first discretize the raw data into some intervals. We can see each interval and its corresponding feature as an item. So the builder can use this item and its label to construct the CBA model.

Discretization method also determines the quality of recognition model. Good discretization method can reduce the model size and save model building time. Details about discretization methods that we use are explained in Sect. 4.

In off-line step, we construct the model. We use discretized data to generate frequent and credible associated rules. The parameter of minimum support is configured 0.025 that can let model size small and accuracy high enough. After generating all qualified rules, we build the classifier model by removing all rules which cover no data.

In on-line step, mobile phone has the information of which are useful features (by feature selection) so that mobile device will only open the sensor that its features are useful. This can reduce the power consumption to log useless data. After collecting user's sensor data, like off-line step, we extract the features from raw data. The classifier model try to match the rules from the model and extracted features to recognize user's activity.

4 Features Extraction and Selection

In the following section, we will explain more about collecting the data inside the smart phone, extract the features, discretize the feature to alter continuous values into categorical type and do feature selection to filter out irrelevant features.

4.1 Data Collection and Features Extraction

In our experiments, we create an android apps to log our activities, based on six activity labels, which are: working, entertainment, sporting, shopping, transportation, and dining. The application records all the sensors data from the smart phone and the corresponding activity label. The apps are being used by 50 peoples who are using android 4.2 smart phones in eight months to collect

Table 2. Sensor and its corresponding features

Sensor	Features	Description
Time	Date	Monday, Tuesday,...to, Sunday
	Period	0–24 h
Accelerometer	AccelAvg	Average force of acceleration
	AccelSd	Standard deviation of force of acceleration
Gyroscope	GyroAvg	Average force of Gyroscope
	GyroSd	Standard deviation of force of Gyroscope
Proximity	GyroSd	Average force of Proximity
GPS	GPSx	Average longitude
	GPSy	Average latitude
	GPSspeed	Average speed
Magnetic	MageAvg	Average Magnetic
	MageSd	Standard deviation of Magnetic
pressure	PressureAvg	Average pressure
Light	LightAvg	Average light

the data. Most users log their activity data for at least one month and the most user logs is about four month worth.

Raw sensor data that have been collected which is recorded every single second, can be divided into four types. First is motion sensor (Accelerometer, Gyroscope, and Proximity) which is designed for collecting user's body motion. Second is location sensor (GPS), which collects user location information. Third is environment sensor (Light, Magnetic and Pressure) which logs the environment data around the user. Final one is time sensor which records the time, containing time of the day and day of the week information. For these sensors data, we extract some statistical features such as average and standard deviation. Each sensor data is put together in one row, grouped together in 10 second frame. Several sensor data is also split based on the axis, such as accelerometer, gyroscope, magnetic and GPS. The details of the sensors and their features are shown in Table 2. After extracting statistical features from original raw data, we focus on the distribution given for every feature. Each feature has unique distribution. Each activity label on each feature also show some unique distribution. In the following section, feature distribution is used to select the feature which fit best to infer the activity label.

4.2 Data Discretization

Traditionally, some classification learning algorithms and feature selection assume that attributes are in numerical values. In this paper, our method is also assuming that input for the model is in numerical values. In order to build activity recognition system for mobile device using Rule-based classifier, we need to discretize the data first, as the data which is collected from mobile phone sensors are in continuous values.

Algorithm 1. LGD (Length-Gini Discretization) algorithm

```
Input: Continuous data to be discretized: $S$
Output: The discrete data: $S'$
Define $BestCut = \emptyset$;
Define $NumberOfInterval = 1$;
while $NumberOfInterval < SizeofS$ do
    Define $Cutway = S$ equal weight partition into $NumberOfInterval$;
    Define $\alpha = 0$ to 1 (suggest 0.3 to 0.6);
    Define $Gini =$ Gini index after partition data $S$;
    Define $L_{value} = NumberOfInterval/Numberofthisdata$;
    Define $LGD_{index} = \alpha * Gini + (1 - \alpha) * L_{value}$;
    if $LGD_index > BestLGD$ then
        Set $BestCut$ to $Cutway$;
        Set $BestLGD$ to $LGD_{index}$;
    end
    Increase $NumberOfInterval$ ;
end
return $Cutway$
```

Our work uses two famous standard (entropy and Gini) to determine where is the best cut point. We do some observation to find how many intervals we need.

By observing the accuracy and model size of CBA under different ways of discretization, the result shows two things. First, cutting few intervals will let each interval blend too much meaning and let this interval useless. Second, although cutting many intervals can let each interval pure, but also it will let that each interval's coverage becomes smaller and lose the opportunity to co-work with other attributes.

Based on observation, we need a standard which consider both number of interval and confusion of each interval. In this paper, we design Interval Length-Gini Discretization (LGD) method which is modified from Gini index. LGD using Gini to consider the confusion of each interval and add a formula to consider number of interval for partitioning. The smaller LGD value is, the better partition is. This algorithm use iterator way to choose the best partition point. Each round raw data will be partitioned into different intervals. Then LGD will calculate the LGD value of this partition. If the LGD value is the smallest one. LGD will record this partition way. After that, number of partition will be increase and go to next round. Algorithm 1 shows the detailed procedure of the LGD. In addition to LGD, we also adopt another discretization method called MDLP [3]. MDLP use the concept of MDL so it determine to cut if and only if information gain is more than the loss to depict the cut point. We implement a recursive algorithm to discretize our sensor data.

Both LGD and MDLP have its own advantages. While MDLP has lower training time, it has lower accuracy performance. Because MDLP uses the concept of MDL, so it accepts data loss. On the other hand, LGD is designed to improve the accuracy of associative classifier so it will have higher accuracy in the cost of having longer training time.

4.3 Feature Selection

In this paper, we use entropy as a measure to calculate the correlation between each feature and class label. As the entropy values gets bigger, feature becomes more impure so the correlation between the feature and the class label become lower. The information about feature impurity shows us that the feature is able to distinguish the class label well or not. For each feature, which are in sensor-value pair, entropy information will be added so we can determine the quality of each feature. We will use one-cut feature selection and iterative-based feature selection, where one-cut only consider one threshold to select the feature, and memory-iteration-based feature selection will use several iterations to select the feature better.

One-cut approach is simply using entropy threshold and remove the feature which has entropy above the threshold. In the different threshold, the result of feature selection may result different output. The ideal situation is where we only select few features but the performance of learning algorithm is remarkably high. For example, selecting entropy 0.5 as the threshold on specific user may remove up to 70 % on the features but is able to only reduce the accuracy by 15 %. In general, one-cut approach has the advantages of having fast and simple implementation. Another approach, memory-iterative-based is using several iteration to select the features by also using locally weighted naive Bayesian [4] as the learning algorithm. The advantage of using Bayesian is that it only requires small amount of training data to estimate the parameters (i.e. mean and variance of the variables) necessary for classification. Because independent variables are assumed, only the variance of the variables for each class needs to be determined. For every iteration the number of features are decreasing as the threshold goes lower or tighter. This process is repeated until entropy threshold reaches minimum entropy value. The process can also stop before entropy threshold reaches minimum level if the overall performance is decreased to some level. In this paper, we use 0 as the minimum entropy value and 80 % of original accuracy as the minimum overall performance requirement. Another issue is that in some particular thresholds, certain selected features may affect the performance of learning algorithm significantly. We also found out that some features can be considered special, as they contradict each other. Feature is having high entropy which is considered as bad feature because they are impure, but they have also many occurrences in the sensor-value pair set. These certain features lead us to use temporary variable inside iteration to obtain better feature subset result. The process can be explained as follow. We remove several features in each entropy threshold. Removed features are first stored in temporary variable first. Selected features will be first evaluated using learning algorithm and then if the performance of learning algorithm is significantly reduced compared to the previous iteration, then the removed features will be added back to feature list and tag them as special feature. The detail memory-iteration-based feature selection is explained in Algorithm 2.

Algorithm 2. One-step-memory iteration-based feature selection

```
Input: Feature sets: $S$, starting threshold $E$, entropy deviation $D$, reduction tolerance $T$
Output: Feature subsets: $S'$
Define $Temporary = \emptyset$;
Set $Acc = 0$;
while $threshold > 0$ do
    foreach $f$ in $S$ do
        if $f.entropy > threshold$ OR $f.tag$ is false then
            Remove feature $f$ from $S$ ;
            Add feature $f$ to $Temporary$ ;
        end
        Evaluate feature subset using learning algorithm ;
        $prevAcc = Acc$ ;
        $Acc$ = current accuracy of learning algorithm ;
        if $prevAcc - Acc > tolerance$ then
            foreach $temp$ in $Temporary$ do
                set $temp$.tag to be true ;
                put back $temp$ to $S$ ;
            end
        end
        $threshold = threshold - deviation$ ;
        Clear $Temporary$ ;
    end
end
```

Table 3. Statistics information of the real dataset

	Average	max	min
Number of day to collect of a user	44	136	7
Collected data size of a user	634 MB	2289 MB	1 MB

5 Experiments

In this section, we evaluate the performance, including accuracy, efficiency and space usage, of the proposed recognition method and compare it with other baseline classifier methods. We also explain the impact of using different discretization methods. Finally, the influence of feature selection is delivered.

5.1 Experimental Environment

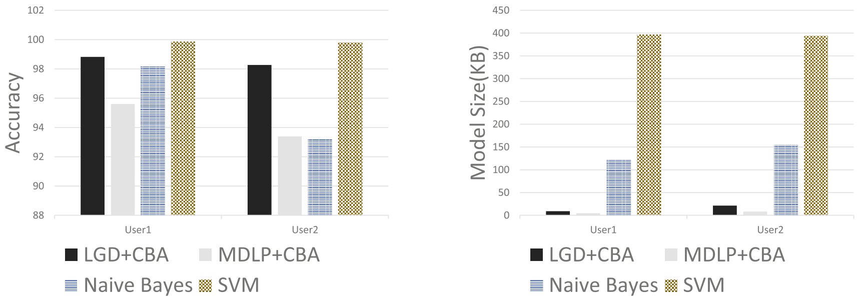
First of all, we introduce the environment of our experiments, including the characteristics of the dataset we used and the measurements to evaluate the recognition performance.

Dataset Description. In this paper, we conduct extensive experiments on real dataset which is collected from volunteers. We implement our logging program on the Android 4.2 platform to record the user's activity and its sensor data. The logger will send the record to server every 10 minutes. For this dataset, we collect 50 participant's activity behavior from May 2013 to December 2013. In our experiments, each users log their activity data for at least one week and the most user log is about four month worth. Each user data is separated as 80 % for training and 20 % for testing. Table 3 shows the statistical information of our dataset.

Compared Methods. Although we introduce some existing activity recognition method, they have different setting with our experiment, such as using different set of sensors and limited equipped position of the mobile device. Therefore, we conduct two famous method, Naive Bayesian and SVM, as our comparison. Naive Bayesian is commonly-used algorithm as it only uses probability model to predict the class label. SVM is another commonly-used algorithm which has high accuracy in most of the experiment cases. We implement these two methods as our comparison. The testing platform and input data is the same as our method. We use WEKA library which provide many famous machine learning algorithm to implement both Naive Bayesian and SVM method. For our method, we also evaluate the performance under different discretizations and feature selections.

5.2 Experimental Results

After we conduct the experiment, we are able to present that our proposed method has an advantages compared to another commonly-used algorithms.



(a) Accuracy Comparison

(b) Model Size Comparison

Fig. 2. Different Recognition Method Comparison where CBA's minimum support = 1 % and SVM and Naive Bayesian use default setting

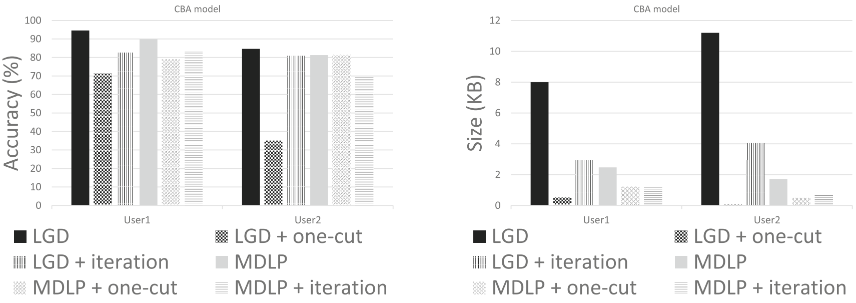
As we concern more to activity recognition in the mobile device, we would desire low model size more, which in some case need to sacrifice some accuracy performance.

Accuracy and Resource Comparison. We first show the experiments result of accuracy on real user dataset. Figure 2(a) show the accuracy performance of our proposed algorithm with two other algorithms. In general, the accuracy of our method are close to SVM and slightly better than Naive Bayesian.

Then, we evaluate the model size on real user dataset. Model size is important issue for mobile. So that the model can be built in sensor hub, model size need small enough. Figure 2(b) show the model size of our proposed algorithm with two other algorithms. For testing the model of SVM and Naive Bayesian, we use WEKA to store the model in the hard drive and remove all default explanation by WEKA. In terms of model size, our proposed method take a runaway lead.

Impact of Features Selection. Figure 3(a) show the influence of accuracy under feature selections and Fig. 3(b) show the influence of model size. As figure above shown, the experiment show that feature selections can reduce half of model size under not reducing too much accuracy. In our observation, if the user has complex lifestyle, memory-iteration-based approach show more stable result, which have smaller performance reduction compared to one-cut approach.

Parameter Studies. Discretization only has one parameter α which balances the gini index and the number of intervals. Table 4 show the model's (i.e. LGD+CBA) accuracy under different α. The value of α should be between 0.3 to 0.6 because (1) it have good accuracy and (2) setting α too low will increase little accuracy but increase much run time. In model building step, there is one parameter called minimum support s. For these parameter, there are much existing research to discussion. In this paper, we suggest that s set to be 1 % to let accuracy higher or set to be 2.5 % to let model size smaller.



(a) Accuracy Comparison (b) Model Size Comparison

Fig. 3. Different Feature Selection Method Comparison where CBA's minimum support = 2.5 %

Table 4. Accuracy Comparison under different α where CBA's minimum support = 2.5 %

α	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
accuracy	50.84	94.76	95.27	94.93	94.59	94.26	94.42	94.42	94.59	93.75	92.91	88.85	89.53	88.01	88.18	83.61	84.63	84.29	84.29	83.61	83.61

6 Conclusion

In this paper we proposed a complete framework ACF for activity recognition using smart phone. In order to deal with continuous value data and to reduce model size while preserving the accuracy performance, we apply two phases of preprocessing: discretization and feature selection. For discretization, we use two approaches, LGD which use Gini index to consider the confusion of each interval and MDLP which use the concept of MDL (Minimum Description Length) to cut if and only if the information gain is more than the loss to depict the cut point. Feature selection technique that we use here including one-cut approach and memory-iteration-based approach. One-cut approach has better running time performance while memory-iteration-based approach has better accuracy performance. In our experimental study, recognition framework that we proposed has also the advantage on smart phone usage as it has smaller model size and considered having high accuracy compared to two other commonly-used algorithms.

Acknowledgement. Wen-Chih Peng was supported in part by the National Science Council, Project No. 102-2221-E-009-171-MY3 and 100-2218-E-009-016-MY3, by hTC and by Academic Sinica Theme project, Project No. AS-102-TPA06.

References

1. Dernbach, S., Das, B., Krishnan, N.C., Thomas, B.L., Cook, D.J.: Simple and complex activity recognition through smart phones. In: IE, pp. 214–221 (2012)

2. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: ICCCN, pp. 65–72 (2005)
3. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: IJCAI, pp. 1022–1029 (1993)
4. Frank, E., Hall, M., Pfahringer, B.: Locally weighted naive bayes. CoRR, abs/1212.2487 (2012)
5. Gini, C.W.: Variability and mutability, contribution to the study of statistical distributions and relations (1912)
6. Honda, D., Sakata, N., Nishida, S.: Activity recognition for risk management with installed sensor in smart and cell phone. In: Jacko, J.A. (ed.) Human-Computer Interaction, Part III, HCII 2011. LNCS, vol. 6763, pp. 230–239. Springer, Heidelberg (2011)
7. Lidstone, R.: Website news: Atmel sensor hub mcu helps enhance samsung galaxy s4 user experience, battery life. MobilityTechzone, May 2013
8. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining, pp. 80–86 (1998)
9. Raileanu, L.E., Stoffel, K.: Theoretical comparison between the gini index and information gain criteria. Ann. Math. Artif. Intell. **41**(1), 77–93 (2004)
10. Rashidi, P., Cook, D.J.: Mining sensor streams for discovering human activity patterns over time. In: ICDM, pp. 431–440 (2010)
11. Ryoo, M.S.: Human activity prediction: Early recognition of ongoing activities from streaming videos. In: ICCV, pp. 1036–1043 (2011)
12. Schüldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local svm approach. ICPR **3**, 32–36 (2004)
13. Stikic, M., Larlus, D., Ebert, S., Schiele, B.: Weakly supervised recognition of daily life activities with wearable sensors. TPAMI **33**(12), 2521–2537 (2011)
14. Zhu, S., Wang, D., Yu, K., Li, T., Gong, Y.: Feature selection for gene expression using model-based entropy. IEEE/ACM Trans. Comput. Biology Bioinform. **7**(1), 25–36 (2010)

Personalized Smartphone Wearing Behavior Analysis

Yen-Hsuan Lin and Yi-Ta Chuang(✉)

Department of Computer Science, National Chiao-Tung University,
1001 University Road, Hsinchu City 30010, Taiwan
izeki.cs96g@nctu.edu.tw

Abstract. Next generation smartphones have the ability to sense user contexts such as mobility, device wearing position, location, activity, emotion, health condition. Many apps utilize user contexts to provide innovative services, e.g., pedometer, advanced navigation and location based services. Two of the most important user contexts are mobility patterns (still and walk) and device wearing positions (hand, arm, chest, waist and thigh). We call these two user contexts "wearing behavior". In this paper, we propose a 3-stage framework to recognize smartphone wearing behaviors by utilizing sensor data from smartphones. The framework starts with data preprocessing to extract sensor features and generate ground truths. After the data preprocessing, a threshold based finite state machine utilizes the sensor features to determine whether the smartphone is attached or not. Finally, a decision tree model is built based on the ground truth to determine the wearing behaviors. The experiment results show that our approach can achieve 94 % accuracy in average.

Keywords: Mobility · Next generation smartphone · User context · Wearing position · Wearing behavior

1 Introduction

Next generation smartphones have the ability to sense user contexts such as mobility, device wearing position, location, activity, emotion, health condition. Many apps utilize user contexts to provide innovative services, e.g., pedometer, advanced navigation and location based services. Two of the most important user contexts are mobility patterns and device wearing positions. For example, an adaptively models could be developed for the pedometer to detect footsteps more accurately; a better positioning method could be established for advanced navigation and location based services. In addition, an adaptive sensor duty cycle policy could be designed based on mobility patterns and wearing positions to support continuously user context sensing. In this work, mobility patterns and wearing positions are called wearing behaviors, and we want to use various sensors embedded in smartphones to detect the wearing behaviors.

There are many embedded sensors in a smartphone, e.g., accelerometer, gyroscope and light sensors. Recently, there are many works using the embedded sensors in a smartphone to detect user contexts [1] and utilize the user context to provide innovative services [2–4]. In [5] and [6], the applications utilize acceleration to detect human

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 318–328, 2014.
DOI: 10.1007/978-3-319-13186-3_30

mobility by the embedded accelerometer in the smartphone. However, these studies only consider the devices wearing in a fixed body position. This constraint restricts the applicability of their methods in the smartphone application since the smartphone users may not place their phones in fixed positions. In this work, we propose to consider both mobility and wearing position as a new user context called the wearing behavior and propose a framework to detect this new user context so that these applications requiring wearing behavior information can be adopted in the smartphone applications.

In this work, the wearing behavior is defined as a set of ordered pairs of mobility pattern set and wearing position set. We consider human mobility such as walking and stationary states, and the smartphone wearing positions including hand, arm, chest, waist and thigh. We develop a 3-stage approach to detect the smartphone wearing behavior. A data collection tool was developed to help us collect sensor data and label wearing behaviors. In the first stage, these labeled data are preprocessed to extract features and generate ground truth. In the second stage, the extracted features are used to build a threshold based finite state machine (FSM) to detect whether a smartphone is attached or not. This stage is reasonable because it is meaningless to detect the wearing behavior if the smartphone is not worn in the body. In the final stage, the extracted features are used to train a decision tree (DT) model for wearing behavior recognition. If it is detected that a smartphone is attached, the decision tree model is used to determine the wearing behavior.

The contributions of this work are several folds. Firstly, we proposed a threshold based FSM by using embedded sensors such as gyroscopes and accelerometers to detect the attachment of smartphones. By this mean, it is possible to improve user experience. For example, if it is detected that the smartphone is not attached, it can automatically tune the volume up so that users may not miss a phone call. Secondly, we developed a framework for personalized wearing behavior recognition. Based on the relationship of sensor readings to wearing positions and mobility, a personalized model is established by data mining techniques for personalized smartphone wearing behavior recognition. With this wearing behavior information, it is very useful to assist in applications, such as pedometer, advanced navigation and location based services, which require personalized usage behaviors. Last but not the least; the extracted features may not be only useful in the wearing behavior analysis but also in other user context related studies.

The rest of this paper is organized as follows. Section 2 reviews some related works on user context applications. In Sect. 3, the overview of our approach is introduced. Section 4 presents the data preprocessing process that extracts features and ground truth. The attachment recognition process that use a threshold based FSM is introduced in Sect. 5. The smartphone wearing behavior recognition process that applied DT is described in Sect. 6. Conclusions are drawn in Sect. 7.

2 Related Works

In this section, we give a briefly review on the related works to the user context related applications and research works.

In [1], a mechanism is proposed to detect user context for mobile and social networking applications. In [2, 3], the authors utilized the user context in the app usage to design energy saving mechanisms. In [4], a fast app launching mechanism is proposed based on the user context. In [5], this application turns the phone screen off automatically when it detects the user puts the phone into a pocket or onto a table and turns the screen on automatically when it detects the user takes the phone out or up. In [6], this application uses the accelerometer to detect whether the phone is on the hand. If it detects the phone is on the hand, it keeps the screen on.

In this work, we consider both mobility and wearing position as a new user context called the wearing behavior and propose a framework to detect this new user context. The detected wearing behavior could be utilized in these applications requiring wearing behavior information so that the user can enjoy these innovative services in smartphones.

3 Proposed Approach

To analyze the wearing behaviors, we propose a two-phase framework. The phases are training phase and inference phase. Each phase is divided into three stages: data preprocessing, attachment recognition and wearing behavior recognition as illustrated in Fig. 1.

For the training phase, in the data preprocessing stage, the sensor data with wearing behavior labels are collected. In this stage, sensor features are extracted from sensor data and the sensor features are input to the attachment recognition stage. In addition, the sensor features with wearing behavior labels are the ground truths and the ground truths are input to the wearing behavior recognition stage. In the attachment recognition stage, the sensor features are used to determine the threshold in the FSM. In the wearing behavior recognition stage, the ground truths are used to build the DT model. We will introduce the sensors features used in our approach in Sect. 4.

For the inference phase, in the data preprocessing stage, sensor features are extracted from sensor data and the sensor features are input to the attachment recognition stage. In the attachment recognition stage, the threshold based FSM determines the attachment state based on the sensor features. The attachment state along with the sensor features are input to the wearing behavior recognition stage. In the wearing behavior recognition stage, if the attachment state is attached, the DT model detects the wearing behaviors. Otherwise, the wearing behavior recognition finishes. We will introduce the threshold based FSM for the attachment recognition and the DT for the wearing behavior recognition in our approach in Sects. 5 and 6, respectively.

4 Data Preprocessing

Recall that our approach comprises the training phase and the inference phase. In order to collect training data for the subsequent model training for the training phase. We develop an application to help us collect sensors data with wearing behavior labels. In this application, users can label the mobility and the wearing positions so that the

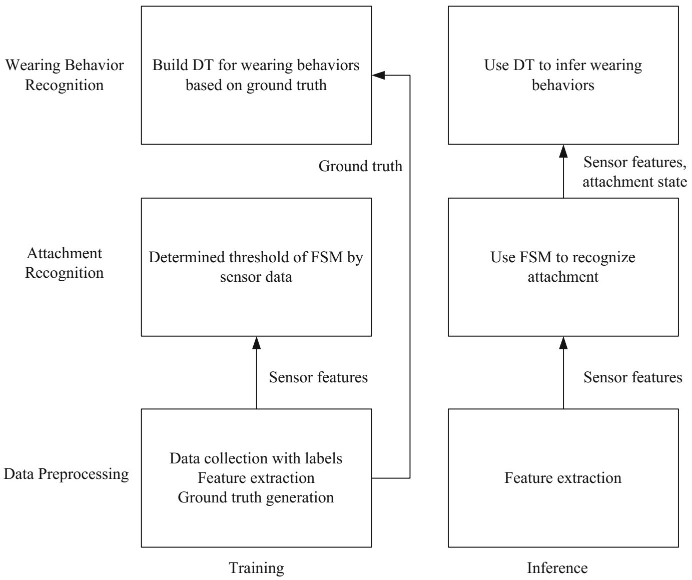

Wearing Behavior Recognition
Build DT for wearing behaviors based on ground truth
Use DT to infer wearing behaviors
Ground truth
Sensor features, attachment state
Attachment Recognition
Determined threshold of FSM by sensor data
Use FSM to recognize attachment
Sensor features
Sensor features
Data Preprocessing
Data collection with labels
Feature extraction
Ground truth generation
Feature extraction
Training
Inference

Fig. 1. Personalized wearing behavior analysis framework.

collected sensor data are with wearing behavior labels. The collected sensor data are preprocessed to extract sensor features. The sensor features are used to determine the threshold for the FSM in the attachment recognition stage. The sensor features with wearing behavior labels are the ground truths and are used to build DT in the wearing behavior recognition stage. For the inference phase, the sensor features are extracted in the data preprocessing stage. After that, the sensor features are input to the attachment recognition stage to determine the attachment state by the FSM. Finally, the sensor features and attachment state are input to the wearing behavior stage to detect the wearing behaviors by the DT model. In our experiment, we use hTC one model to collect sensor data and the sampling rate is 100 Hz.

The wearing behavior considered in this work is an order pair of the mobility patterns and the wearing positions. The mobility patterns in this work include still and walk states. We say the mobility is in still state if the footsteps are no more than two per second. Otherwise, we say the mobility is in walk state. The wearing positions in this work are hand, arm, chest, waist and thigh. The hand position is self-explained; the arm position is that the smartphone is strapped in an armband; the chest position is that the smartphone is placed in a breast pocket; the waist position is that the smartphone is placed in the purse near the waist; the thigh position is that the smartphone is placed in the pocket of trousers.

The sensors and features used in this work are depicted in Table 1. We collect sensors data from the embedded accelerometer, gyroscope and light sensor. The accelerometer and gyroscope are used to detect the movement and direction of the smartphone. The light sensor is used to detect the intensity of light. The intensity of light could be used to determine whether a smartphone is in a pocket. Since the raw sensor data are diverse across different kind of domains, we use several feature extraction technique such as statistic and frequency domain transformation on the sensors data.

Table 1. Sensors and features used in this work.

Hardware sensor	Feature	Unit
Accelerometer	Avg intensity	m/s^2
	Avg intensity of vertical component	m/s^2
	Avg abs intensity of vertical component	m/s^2
	Avg intensity of horizontal component	m/s^2
	Standard deviation	m/s^2
	DFT of vertical components	–
	Phone direction	–
Gyroscope	Avg intensity	rad/s
	Standard deviation	rad/s
Light	Avg intensity	lux

4.1 Sensor Features

This section introduces the sensor features used in our framework. There are 7 features extracted from the accelerometer, 2 features from the gyroscope and 1 feature from the light sensor.

The features extracted from the accelerometer are average intensity of acceleration, average intensity of vertical components of acceleration, average intensity of horizontal components of acceleration, standard deviation of acceleration, Discrete Fourier Transformation (DFT) of vertical components of acceleration and phone direction.

Average intensity of accelerometer ($\overline{\|A\|}$) is calculated by

$$\overline{\|A\|} = \frac{1}{n}\sum\nolimits_{i=1}^{n}\|\boldsymbol{a}_i\|, \tag{1}$$

where $\boldsymbol{a_i}$ is the ith acceleration and n is the number of data in a minute. $\|\ \|$ denotes the norm operation on a vector.

Average intensity of vertical components of acceleration ($\overline{\|A^{\perp}\|}$) is calculated by

$$\overline{\|A^{\perp}\|} = \frac{1}{n}\sum\nolimits_{i=1}^{n}\|\boldsymbol{a}_i^{\perp}\|, \tag{2}$$

where $\boldsymbol{G}_i$ is the gravity of the ith data, $\boldsymbol{a}_i^{\perp}$ is the vertical component of the ith acceleration which is calculated by $\frac{\boldsymbol{a}_i \cdot \boldsymbol{G}_i}{\|\boldsymbol{G}_i\|}\frac{\boldsymbol{G}_i}{\|\boldsymbol{G}_i\|}$ and $\cdot$ denotes the dot product operation on vectors.

Average absolute intensity of vertical components of acceleration ($\overline{|A^{\perp}|}$) is calculated by

$$\overline{|A|^{\perp}} = \frac{1}{n}\sum\nolimits_{i=1}^{n}\left|\frac{\boldsymbol{a}_i \cdot \boldsymbol{G}_i}{\|\boldsymbol{G}_i\|}\right| \tag{3}$$

Average intensity of horizontal components of acceleration ($\overline{\|A^{=}\|}$) is calculated by

$$\overline{\|A^{=}\|} = \frac{1}{n}\sum\nolimits_{i=1}^{n}\|\boldsymbol{a}_i - \boldsymbol{a}_i^{\perp}\| \tag{4}$$

Standard deviation of acceleration (A_σ) is calculated by

$$A_\sigma = \left(\frac{1}{n}\sum\nolimits_{i=1}^{n}\left(\|\boldsymbol{a}_i\| - \overline{\|\boldsymbol{A}\|}\right)^2\right)^{1/2} \tag{5}$$

The magnitude of vertical components of acceleration after DFT at 1 Hz (A_1) is calculated by

$$A_1 = \left\|\sum\nolimits_{j=0}^{2n}\left\|\boldsymbol{a}_j^{\perp}\right\| * e^{-i2\pi j/n}\right\| \tag{6}$$

Phone direction (P) is determined by

$$\mathrm{P} = \left\{\begin{array}{l} 0,\ \text{if}\ \max\{\bar{\boldsymbol{A}}_x, \bar{\boldsymbol{A}}_y, \bar{\boldsymbol{A}}_z\} == \bar{\boldsymbol{A}}_x \\ 1,\ \text{if}\ \max\{\bar{\boldsymbol{A}}_x, \bar{\boldsymbol{A}}_y, \bar{\boldsymbol{A}}_z\} == \bar{\boldsymbol{A}}_y \\ 2,\ \text{if}\ \max\{\bar{\boldsymbol{A}}_x, \bar{\boldsymbol{A}}_y, \bar{\boldsymbol{A}}_z\} == \bar{\boldsymbol{A}}_z \end{array}\right\}, \tag{7}$$

where $\bar{\boldsymbol{A}}_x$, $\bar{\boldsymbol{A}}_y$ and $\bar{\boldsymbol{A}}_z$ denote the average intensity of acceleration in x, y and z directions, respectively.

Average intensity of gyroscope ($\bar{G}$) is calculated by

$$\bar{G} = \frac{1}{n}\sum\nolimits_{i=1}^{n}\|g_i\|, \tag{8}$$

where g_i is the ith angular velocity.

Standard deviation of gyroscope (G_σ) is calculated by

$$G_\sigma = \left(\frac{1}{n}\sum\nolimits_{i=1}^{n}\left(\|\boldsymbol{g}_i\| - \|\bar{\boldsymbol{G}}\|\right)^2\right)^{1/2} \tag{9}$$

Average intensity of light ($\bar{\boldsymbol{L}}$) is calculated by

$$\bar{L} = \log\left(\frac{1}{n}\sum\nolimits_{i=1}^{n}\|L_i\|\right), \tag{10}$$

where L_i is the ith light luminance.

5 Attachment Recognition

In the attachment recognition stage, for the training phase, the sensors features are used to determine the threshold of the FSM. For the inference phase, the threshold based FSM is designed to determine whether the smartphone is attached.

Since the accelerometer and gyroscope are able to measure the human mobility, it is possible to use the long-term variations of acceleration and rotation to determine whether the smartphone is attached. We use A_{th} and G_{th} as the acceleration and rotation thresholds, respectively. A_{th} is determined by

$$A_{th} = \max_{1 < i < n} A_{\sigma i}, \tag{11}$$

where $A_{\sigma i}$ is the ith A_{σ} in the unattached data and n is the number of unattached data. G_{th} is determined by

$$G_{th} = \max_{1 < i < n} G_{\sigma i}, \tag{12}$$

where $G_{\sigma i}$ is the ith G_{σ} in the unattached data.

Intuitively, we use the maximum of A_{σ} and the maximum of G_{σ} from the training data labeled with unattached state as the thresholds. Based on the thresholds, the attachment state can be determined by

$$\text{State} = \left\{ \begin{array}{ll} \text{unattached,} & \text{if } A_{\sigma} < T_A \textit{ and } G_{\sigma} < T_G \\ \text{attached,} & \text{otherwise.} \end{array} \right\} \tag{13}$$

However, this method may cause false detection. To resolve the false detection and improve the detection accuracy, a two-bit FSM is designed.

There are four states in the FSM: A/A, U/A, A/U and U/U. Each state uses two bits to record the previous and current attachment state. The character ‘A’ means “attached” and the character ‘U’ means “unattached”. For example, U/A state means the previous attachment state is unattached and the current attachment state is attached. The state transition moves as follows. On the A/A state, when the attachment state is unattached, the state transits to the A/U state; when the attachment state is attached, no state transition occurs. On the U/A state, when the attachment state is unattached, the state transits to the A/U state; when the attachment state is attached, the state transits to the A/A state and the FSM reports the smartphone is attached. On the A/U state, when the attachment state is unattached, the state transits to the U/U state and the FSM reports the smartphone is unattached; when the attachment state is attached, the state transits to the U/A state. On the U/U state, when the attachment state is unattached, no state transition occurs; when the attachment state is attached, the state transits to the U/A state. Figure 2 illustrates the state transition of the FSM.

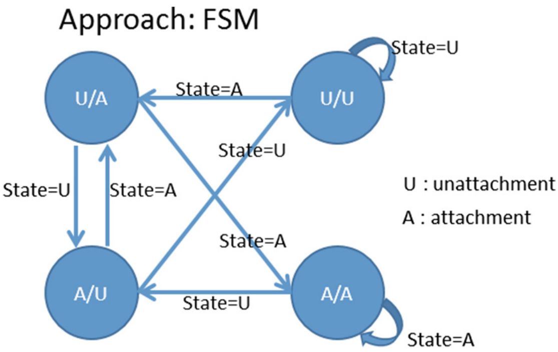


Fig. 2. The state transition diagram of the finite state machine.

6 Wearing Behavior Recognition

The final stage in our framework is the wearing behavior recognition stage. For the training phase, the ground truths generated in the data preprocessing stage are used to train the DT model. We use the REPTree algorithm [7] to build the DT model by the Weka data mining tool [8]. For the inference phase, the feature sets and attachment states from the attachment recognition stage are input to the DT model, and the wearing behavior is determined by the DT model. Figure 3 shows the process concept in the wearing behavior recognition stage.

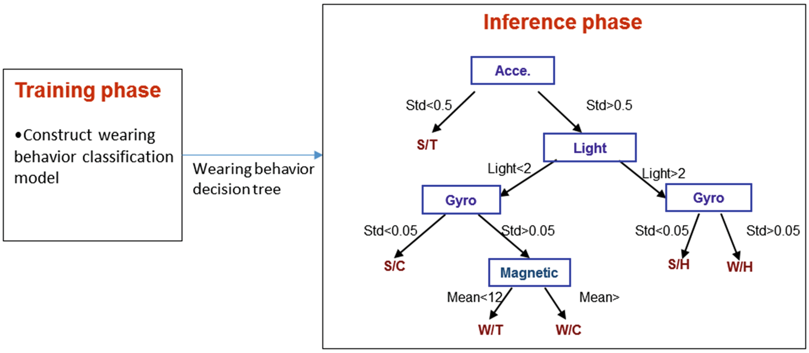


Fig. 3. The process concept in the wearing behavior recognition stage.

In this work, the REPTree algorithm is used to build the DT model for wearing behavior classification. The REPTree algorithm utilizes information gain/variance to build a decision/regression tree and prunes the tree by using reduced-error pruning (with backfitting). The REPTree algorithm uses the information gain [9] to create nodes in the decision tree. Let T denote the set of training data. The training data, i.e., the ground truths in this work, is in the form $X = (x_1, x_2, x_3, \ldots, x_k, y)$ where $x_i = val(a_i)$ is the value of the ith attribute and y is the wearing behavior label (class label). The entropy of the training data set T is calculated by

$$H(T) = -\sum\nolimits_{i=1}^{n} P(c_i)log_2(P(c_i)) \tag{14}$$

where n is the number of classes and $P(c_i) = \frac{|\{X \in T|y=c_i\}|}{|T|}$ is the probability of the class c_i.

The conditional entropy of training data set T given a value of a feature $x_a = v$ is calculated by

$$H(\{X \in T|x_a = v\}) = -\sum\nolimits_{i=1}^{n} P\{c_i|x_a = v\}log_2(P\{c_i|x_a = v\}) \tag{15}$$

The information gain of the training data set T for an attribute a is defined in terms of entropy.

$$IG(T, a) = H(T) - H(T|a), \tag{16}$$

where $H(T|a) = \sum\nolimits_{v \in val(a)} \frac{|\{X \in T|x_a=v\}|}{|T|} \cdot H(\{X \in T|x_a = v\})$ is the average conditional entropy given the feature a.

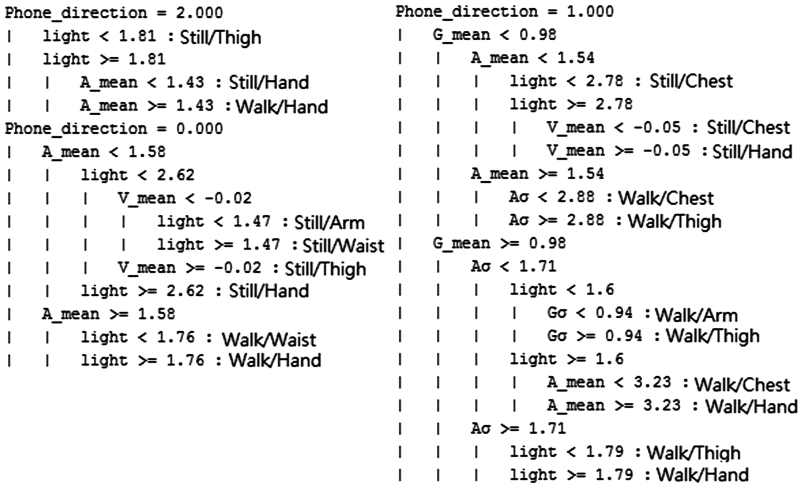
The pseudo code of the REPTree algorithm is depicted in Fig. 4. Note that in line 4 of the pseudo code. If the attribute is numeric type, we need to find the split point to facilitate the calculation of information gain.

```
Build_Tree($T$, $a_{split}$){
   Calculate $IG(T, a)$ for each attribute $a$.
   Find the split point if the attribute is numeric type.
   Find the split attribute $a_{max}$ with the maximum $IG$ among the attributes.
   if $IG(T, a_{max}) > IG(T, a_{split})$ {
      For all $v \in val(a_{max})$ {
         $T = \{X \in T | x_{a_{max}} = v\}$.
         Build_Tree($T$, $a_{max}$).
      }
   }
}
```

Fig. 4. Pseudo code of building decision tree

Two users use the data collection application to collect around 500 training data for each wearing behaviors. Figure 5 is the DT model built from a user dataset by Weka. The maximum depth of tree was set to five to avoid overfitting. From the result of the DT model, we observed that three features: phone direction, light intensity and average intensity of acceleration were the three significant features in the classification.

The phone direction and light intensity were very useful to detect the wearing positions. The phone direction is usually the same for different wearing positions. For example, the phone direction is usually perpendicular to the ground when the smartphone is placed in the armband; the phone direction is usually parallel to the ground when the smartphone is placed in the purse near the waist. The light intensity is useful to detect the position where the difference in luminosity is obvious, e.g., the purse near the waist and the pocket of trousers. The average intensity of acceleration was useful to detect the mobility.



```
Phone_direction = 2.000
|   light < 1.81 : Still/Thigh
|   light >= 1.81
|   |   A_mean < 1.43 : Still/Hand
|   |   A_mean >= 1.43 : Walk/Hand
Phone_direction = 0.000
|   A_mean < 1.58
|   |   light < 2.62
|   |   |   V_mean < -0.02
|   |   |   |   light < 1.47 : Still/Arm
|   |   |   |   light >= 1.47 : Still/Waist
|   |   |   V_mean >= -0.02 : Still/Thigh
|   |   light >= 2.62 : Still/Hand
|   A_mean >= 1.58
|   |   light < 1.76 : Walk/Waist
|   |   light >= 1.76 : Walk/Hand
```

```
Phone_direction = 1.000
|   G_mean < 0.98
|   |   A_mean < 1.54
|   |   |   light < 2.78 : Still/Chest
|   |   |   light >= 2.78
|   |   |   |   V_mean < -0.05 : Still/Chest
|   |   |   |   V_mean >= -0.05 : Still/Hand
|   |   A_mean >= 1.54
|   |   |   Aσ < 2.88 : Walk/Chest
|   |   |   Aσ >= 2.88 : Walk/Thigh
|   G_mean >= 0.98
|   |   Aσ < 1.71
|   |   |   light < 1.6
|   |   |   |   Gσ < 0.94 : Walk/Arm
|   |   |   |   Gσ >= 0.94 : Walk/Thigh
|   |   |   light >= 1.6
|   |   |   |   A_mean < 3.23 : Walk/Chest
|   |   |   |   A_mean >= 3.23 : Walk/Hand
|   |   Aσ >= 1.71
|   |   |   light < 1.79 : Walk/Thigh
|   |   |   light >= 1.79 : Walk/Hand
```

Fig. 5. The result of the decision tree build by the Weka REPTree algorithm.

We used the same data collected from the users to evaluate the DT model by Weka. In the experiment, 80 percentage of the dataset was used as training set to train the DT model, and the rest 20 percentage of the dataset was used to test the DT model. Table 2. is the evaluation result. The accuracy was evaluated in terms of true positive (TP) rate, false positive (FP) rate and precision. The true positive is defined as the number of correct classification and the false positive is defined as the number of incorrect classification. The precision is defined as the true positive over the sum of the true positive and the false positive. The Precision, TP rate and FP rate were 93.9 %, 93 % and 1.3% in average, respectively. We can observe that the TP rates of still/waist and walk/arm were not good (34.5 % and 62.5 %, respectively). The still/waist was usually detected to the still/thigh. This is because the phone direction was usually the same when the user is sitting and the smartphone is placed in the purse near the waist or the pocket of trousers. Similar observation could be discovered in the walk/arm and walk/thigh. The phone direction is similar when the user is walking and the smartphone is strapped in the armband or the pocket of trousers. Nevertheless, the proposed approach is able to detect the wearing behaviors in most cases.

Table 2. The accuracy of the wearing behavior recognition.

Class	TP rate	FP rate	Precision
S/H	0.86	0	1
S/A	1	0.004	0.818
S/C	1	0.02	0.864
S/W	0.345	0	1
S/T	1	0.045	0.827
W/H	1	0	1
W/A	0.625	0.002	0.833
W/C	1	0.002	0.978
W/W	1	0	1
W/T	0.981	0.009	0.972
Weighted Avg	0,93	0,013	0.939

7 Conclusion

In this paper, we proposed a 3-stage framework to detect the wearing behaviors. The wearing behavior is a combination of human mobility and wearing positions. The framework consists of the training phase and the inference phase. Our wearing behavior detection framework starts with the data preprocessing to collect sensor data, extract sensor features and generate ground truths. The sensor features and ground truths are used to train thresholds and classification model in the attachment recognition stage and the wearing behavior recognition stage. After the models are ready, the framework can detect the wearing behaviors based on the sensor features. We evaluate our approach by the weka data mining tool on sensor data from 2 users. The experiment result shows that our approach can achieve the precision at 94 % in average. In the future, we will extend our work on more mobility patterns and wearing positions.

References

1. Santos, A.C., Cardoso, J.M.P., Ferreira, D.R., Diniz, P.C., Chaínho, P.: Providing user context for mobile and social networking applications. Perv. Mobile Comput. **6**(3), 324–341 (2010)
2. Pathak, A., Jindal, A., Hu, Y., Midkiff, S.: What is keeping my phone awake? Characterizing and detecting no-sleep energy bugs in smartphone apps. In: Mobisys (2012)
3. Nath, S.: ACE: exploiting correlation for energy-efficient and continuous context sensing. In: Mobisys (2012)
4. Yan, T., Chu, D., Ganesan, D., Kansal, A., Liu, J.: Fast app launching for mobile devices using predictive user context. In: Mobisys (2012)
5. https://play.google.com/store/apps/details?id=com.plexnor.gravityscreenoffpro
6. https://play.google.com/store/apps/details?id=net.vmid.bettersleep&hl=zh_TW
7. http://wiki.pentaho.com/display/DATAMINING/REPTree
8. http://www.cs.waikato.ac.nz/ml/weka/
9. http://en.wikipedia.org/wiki/Information_gain_in_decision_trees

Activity Recognition Using Discriminant Sequence Patterns in Smart Environments

Been-Chian Chien(✉) and Rong-Sing Huang

Department of Computer Science and Information Engineering,
National University of Tainan, 33, Sec. 2, Su-Lin St., Tainan 70005,
Taiwan, Republic of China
bcchien@mail.nutn.edu.tw

Abstract. Smart environment is one of the important research issues in the area of ambient intelligence and pervasive computing. The high accurate activity recognition is the basis of supporting high-quality service for users in smart environments. In practical applications, the detected signals of monitoring smart space generally come from multiple heterogeneous sensors. Since the streaming data generated by multi-sensor are real time, continuous and noisy, recognizing activities accurately in daily living space is a difficult task. This paper proposed a novel activity recognition scheme for multi-sensor streaming data based on discriminant sequence patterns and activity transition patterns. The efficient pattern mining methods and effective activity predicting algorithms are developed for activity recognition. The experiments apply two well-known datasets, WSU and Kasteren datasets, to verify the performance of the proposed methods. The results show that the models based on the proposed discriminant sequence patterns gain effective recognition rates in both metrics of time-slide activity accuracy and class activity accuracy in comparison with HMM on incremental learning of on-line recognition paradigm.

1 Introduction

One of recent researches on pervasive computing is to integrate the technologies of sensors and machine learning to develop smart environments for human living. A smart environment is usually equipped with different sensors in the living space. The objective is to detect the environmental status and users' activities for providing proper services to human daily life. The satisfaction of a user's intention and request is usually used to assess the success of a smart environment. Generally, a high-accurate activity recognition scheme can increase the reliability of examining users' intention so as to improve the service quality of a smart environment. Hence, recognizing activities accurately with multiple sensors is one of the critical tasks in the research of smart environments.

For detecting users' activities in a smart environment, different sensors are usually distributed around the target space. The sensing data from multiple sensors generate a collection of multi-dimensional streaming data. Activity recognition in a smart environment can be treated as the sequence classification problem on multi-sensor data streams within a specific period of time. Several challenges exist in this issue: First of

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 329–341, 2014.
DOI: 10.1007/978-3-319-13186-3_31

all the noise of operating sensors in the environment cannot be avoided. The signal interference often occurs among sensors due to abnormal awareness or users' unintentional behavior. The second obstacle is that data sequences may contain partial duplicate subsequences among distinct activities. Because the sensing data are received on-line and real-time, it is hard to separate the sensing data sequences into individual activity fragments precisely. Due to the above problems, it is difficult for researchers to find a general model to resolve the issue of activity recognition.

Most activity recognition methods proposed in the last decade are based on Hidden Markov Model (HMM) and Conditional Random Fields (CRF), such as [1–4]. The traditional HMM models use maximum likelihood to find the best parameters for the model based on a set of classified sequential data. The previous results also show that the HMM model performs well in off-line circumstance of activity training and labeling. However, as the identification task are employed in on-line multi-sensor data streams, classification models may not be able to be re-trained completely in time for catching up the last formulating model.

In this paper, we propose a novel activity recognition scheme based on discriminant sequence patterns and activity transition patterns for on-line detecting and incremental learning in sequential data streams. We first propose efficient mining approaches to generate discriminant sequence patterns and analyze activity transition patterns. Then, the activity predicting methods and classification algorithms are developed to recognize activities from a sequence of multi-sensor data stream. The evaluation and experiments are tested on two well known data sets, WSU [5] and Kasteren [2]. The experimental results show that the proposed schemes can improve the effectiveness of both the time-slice accuracy and class accuracy in comparison with HMM.

The rest of this paper is organized as follows. In Sect. 2, the related researches on sequence classification techniques for activity recognition are reviewed briefly. The proposed methods include mining discriminant sequence patterns, analyzing activity transition patterns and the predicting algorithm for activity recognition are presented in Sect. 3. Section 4 gives evaluation of the proposed approaches and discussion the experimental results. Concluding remarks and further work are listed in Sect. 5.

2 Review of Sequential Data Classification

The sequential data classification is one of the primitive techniques of recognizing activity in streaming data. We give brief review and discuss by dividing the sequence classification methods into three categories: model based classification, feature based classification, and distance based classification [6].

The model based sequence classification method assumes that sequences in a class are generated by an underlying generative probabilistic model. The model is usually defined on a specific probability distribution described by a set of parameters. The objective of training phase is to learn the optimized parameters of the model. Then, the classification phase can assign a unknown sequence to the class with highest likelihood or probability. The most common models include Naïve Bayes classifier, Markov Model (MM), Hidden Markov Model (HMM), and Conditional Random Fields (CRF). Naïve Bayes is usually applied to applications which the sequences are independent of each other, e.g. text classification [7]. The MM and the HMM are generally used to model the classification tasks having dependence among elements in their sequences. For example, Yakhnenko et al. apply a K-order Markov Model to classify protein and text sequences [4]. Kasteren et al. apply HMM to recognize activities in smart environment [2].

The feature based classification method generally needs transform sequential data into a multi-dimensional feature vector first before a classification learner, such as decision trees, neural network, SVM, etc., being adopted. Since machine learning algorithms are applied in the classification phase, the extracting meaningful patterns and selecting effective features from sequential data become the most important task. Each extracted features must be short subsequences which satisfy the following rules: (1) It is frequent. (2) It should be distinctive at least one class. (3) It cannot contain redundant features. Chuzhanova et al. apply *k*-grams to generate all possible subsequence in training set, and apply Gamma test to select more informative feature set [8]. Lest et al. propose an Apriori based feature mining method to find distinguished feature patterns [9]. After features being selected, Winnow [10] and Naïve Bayes classifier are employed to classify sequences. Huang et al. proposed a probabilistic based classification algorithm [11] to recognize activities in a smart space based on minimal distinguishing subsequences [12].

The distance based classification method must define a distance function to measure the similarity between two sequences. After computing the similarity between a pair of sequences, some distance based classifiers, like *k* nearest neighbor classifier (KNN) and SVM with local alignment kernel, are used to classify sequence data. Hence, the measuring function of distance is the critical part of the distance based sequence classification. For classifying time series, Euclidean distance and the dynamic time warping distance (DTW) [13] are adopted widely in various applications. For symbolic sequences, the alignment based distance measures are usually adopted [14]. Many variants based on the alignment method are also developed, such as global alignment, local alignment, and region alignment.

3 Activity Recognition Based on Discriminant Patterns

3.1 The Problem and Notation

The activity recognition in multi-sensor environment can be formalized as follows. Let s_i be a multi-sensor data vector and $S = s_1 s_2 \ldots s_t$ is a multi-sensor streaming data sequence. The s_t is the last data of the sequence S. For a finite set of activity classes $C = \{C_1, C_2, \ldots, C_k\}$, C_k is the class label of *k*th activity, $1 \leq k \leq K$. The labeled sequence data are represented as (s_i, C_k), where $1 \leq i \leq t$ and $1 \leq k \leq K$. A sequence S containing s_i having the same activity label in its neighbors will partition the streaming data into activity fragments denoted as $(S_l,\ C_k)$. The S_l be a subsequence fragment of S labeled by the activity C_k, and $|S_l|$ is the length of the subsequence S_l.

Let $D = \{(S_l, C_k) \mid C_k \in C,\ \text{ for } 1 \leq l \leq n, 1 \leq k \leq K\}$ be the training set of activity sequences, where n is the total number of sequence fragments in $\boldsymbol{D}$. The problem of activity recognition on the data streams is to construct an effective recognizer to classify on-line multi-sensor sequences based on the set of labeled training activity sequence ***D***. The objective of activity recognition on data streams needs not only classify the individual sequence data accurately but also partition the streaming sequence effectively.

3.2 Analyses of Discriminant Sequence Patterns

To recognize user activities effectively, the sequence patterns with high discrimination on classifying activities should be found out first. For a sequence of sensor data gathered in continuous time slices, the current user activity is generally dependent on the present and previous sensor signals. For mining discriminant sequence patterns to decide the user activity in current time slice, the mining procedure containing the following two processes is developed.

- Find prefix patterns of each activity by compressed reverse suffix tree.
- Analyze and compute the discriminant sequence patterns.

First, we would like to find the prefix patterns for each sequence fragment of activity. Let $S_l = s_1 s_2 \ldots s_t$ be a sequence fragment for one of the activities in the training set ***D***. The prefix patterns at the time slice *t* are defined as the all possible prefix substrings of s_t. To prevent generating too many dummy patterns, a maximal prefix length (MLP) is set to restrict the length of prefix strings. With a user-defined MPL, a compressed reverse suffix tree is constructed from the sequence fragment S_l for the activity. A compressed reverse suffix tree extracts and stores the possible suffixes of the prefix pattern with maximal prefix length and their frequencies in a compressed form of reverse order. The compressed reverse suffix tree is demonstrated by the following example.

Example 1. Let S_l = ACBCBACA be a sequence fragment of an activity, the compressed reverse suffix tree with maximal prefix length MPL = 3 for all time slice in S_l is created as Fig. 1. The set of prefix patterns S_l are {A, B, C, AC, CB, BC, BA, CA, ACB, CBC, BCB, CBA, BAC, ACA}. Patterns and their corresponding frequencies can be read reversely by the data node beginning with the root from the reverse suffix tree. For example, in Fig. 1, the next level (level 1) of root (level 0) shows that patterns A, B, C repeat 3, 2, and 3 times, respectively. The level 2 depicts the patterns CA, BA, CB, AC, BC and their frequencies are 1, 1, 2, 2, 1, respectively.

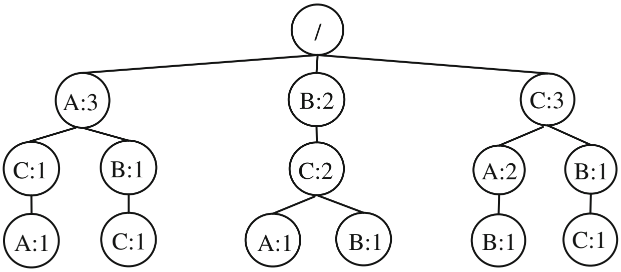


Fig. 1. A compressed reverse suffix tree for

After all the prefix patterns of each activity being counted, the discriminant coefficient [15] is used to analyze and compute the discriminant sequence patterns.

Let P be the set of all found prefix patterns from the training set $\boldsymbol{D}$, and x_{jl} be the frequency of the pattern p_j appearing in the sequence S_l, for $p_j \in P$ and $S_l \in D$. We also define $p_j^{'}$ to be a *super prefix pattern* of pattern p_j if p_j is a proper suffix of $p_j^{'}$.

The set of super prefix patterns for the pattern p_j appearing in the same sequence S_l, H_{jl}, is given as

$$H_{jl} = \left\{ p_j' | p_j' \,\text{is a super prefix pattern of}\; p_j \,\text{and}\, p_j,\, p_j' \,\text{are in}\, S_l \right\}.$$

H_{jl} can be easily derived using the compressed reverse suffix tree along the paths of the pattern p_j to the leaves. Let $\left\|H_{jl}\right\|$ denote the total frequency of the patterns in H_{jl} for S_l. The frequencies of the nodes on the paths are summed up to $\left\|H_{jl}\right\|$.

From x_{jl}, we define the relative importance of the pattern p_j appearing in the sequence S_l to be

$$x_{jl}' = \begin{cases} x_{jl}, & \text{if } \;||H_{jl}|| = 0; \\ x_{jl}/||H_{jl}||, & \text{if } \;||H_{jl}|| \neq 0. \end{cases} \tag{1}$$

Then, w_{jk}, the weight of pattern p_j on the activity class C_k, is defined as

$$w_{jk} = \frac{1}{|C_k|} \sum_{\forall (S_l, C_k)} x_{jl}', \tag{2}$$

where $|C_k|$ is the number of sequence data labeled as C_k in training set $\boldsymbol{D}$. We use the maximum weight of pattern p_j on the activity C_k to normalize w_{jk} as $\tilde{w}_{jk}$ in [0, 1],

$$\tilde{w}_{jk} = \frac{w_{jk}}{\max\limits_{1 \le i \le |C|} \{w_{ji}\}}. \tag{3}$$

The discriminant coefficient of sequence pattern p_j on the activity C_k is defined as

$$d_{jk} = \begin{cases} 0 & \text{if } \tilde{w}_{jk} = 0, \\ \prod\limits_{1 \le i \le |C| i \neq k} |\tilde{w}_{jk} - \tilde{w}_{ji}| & \text{if } \tilde{w}_{jk} \neq 0. \end{cases} \tag{4}$$

The importance of the discriminant sequence pattern p_j on the activity class C_k, z_{jk}, is computed by the following formula:

$$z_{jk} = \log d_{jk} \times \tilde{w}_{jk}. \tag{5}$$

3.3 Activity Transition Patterns

In addition to the sequence patterns, the sequence of users' activities is also an important factor of deciding the current activity of a user in the smart space. In some extreme cases, the predicted activity with the highest discriminant sequence pattern appears in unreasonable order. For example, a user never 'goes to bed' before 'dinner' in the training examples. Hence, it may cause wrong recognition if the 'dinner' activity was detected after the activity 'goes to bed.' The goal of analyzing activity transition patterns is to realize the possible sequence of users' activities and avoid the meaningless activity transition.

Let $S = s_1 s_2 \ldots s_t$ be the streaming data as previous definition, the labeled streaming data are represented as (s_i, c_i), $c_i \in C$, $1 \leq i \leq t$. The set of activity transition patterns T are analyzed and collected as the following definition:

$$T = \{(c_{i-1}, c_i) \,|\, \text{for any } (s_i, c_i), 1 \leq i \leq t, c_i \in C, \text{ and } c_{i-1} \neq c_i\}.$$

The procedure of finding the set of next possible candidate activities for the last data s_t are listed in Fig. 2.

```
Algorithm: Find_candidate
Input: unlabeled data $s_t$, labeled data $s_{t-1}$ with $c_{t-1}$;
Output: the set of the candidate activities $A$;
{
  $A = \{c_i \mid (c_{t-1}, c_i) \in T, c_i \in C\} \cup \{c_{t-1}\}$;
  if ($|A|$ == 1)
    $A = C$;
  endif
}
```

Fig. 2. The algorithm of finding possible candidate activities.

3.4 Sequence Classification and Activity Recognition Algorithm

After mining discriminant sequence patterns and activity transition patterns, we develop the sequence classification methods and activity recognition algorithm based on the two sequence patterns for on-line multi-sensor streaming data.

Assume that $S = s_1 s_2 \ldots s_t$ be the multi-sensor data stream and s_t is the last signal we received and recognized. Let L be the maximal prefix length of the discriminant sequence patterns. The procedure of the sequence classification contains three main stages.

Stage 1. The first stage uses the discriminant sequence patterns with $z_{jk} = 1$ to identify the activity C_k directly if one of the discriminant sequence patterns p_j containing the datum s_t. In this stage, the sequence will be looked forward until the pattern length is L. If the s_i could not decide the activity at this stage, the second stage and the third stage are proceeded.

Stage 2. This stage handles the case of idle signals in which the $s_i, s_{i+1}, \ldots, s_t$ are the same. While the stream data keep the same data, it means that the user is idle. At this moment, the last identified activity label at the previous decision will be assigned to the present signal s_t.

Stage 3. In the third stage, the activity estimation functions are designed to decide the activity for the unlabelled stream data. Let P be the discriminant sequence patterns obtained from the training set $\boldsymbol{D}$. For a stream data S, if s_t is the last unlabeled multi-sensor data. We first find the set of possible super prefix patterns of s_t with maximal length L in S, $s_{(t-l+1)} \ldots s_t$ of length l, $1 \le l \le L$. Next, we search the super prefix patterns in P, the discriminant sequence patterns obtained from the training set $\boldsymbol{D}$, to find the corresponding importance of the discriminant sequence pattern. Then, two activity estimation functions, average discrimination μ_k^{Ave} and maximum discrimination μ_k^{Max} are given to measure the membership of s_t belonging to each activity class:

$$\mu_k^{Ave}(s_t) = \frac{\sum_{l=1}^{L} \left[l \times \frac{z_{jk}^l}{Z_j} \times \frac{z_{jk}^l}{N_k} \right]}{\sum_{l=1}^{L} l}, Z_j = \sum_{k=1}^{K} z_{jk}^l, N_k = \sum_{(S_l, C_k) \in \boldsymbol{D}} |S_l|; \tag{6}$$

$$\mu_k^{Max}(s_t) = \max_{1 \le l \le L} z_{jk}^l; \tag{7}$$

where z_{jk}^l is the importance of the discriminant sequence pattern p_j with length l, and $|S_l|$ is the length of the sequence S_l, for $1 \le k \le K$.

Before deciding the activity of s_t, the set of candidate activities A should be found by the algorithm *Find_candidate* in Sect. 3.3. At last, The activity of s_t is assigned to C_k, where

$$k = \arg_i \max_{C_i \in A} \{\mu_i^{Ave}(s_t)\} \text{ or } k = \arg_i \max_{C_i \in A} \{\mu_i^{Max}(s_t)\}.$$

The detailed activity recognition algorithm is listed in Fig. 3. The three important variables in the algorithm are depicted as follows: (1) *pos* is the index of the signal needed to be recognized. (2) *un_pos* is the index of starting unlabeled signal. (3) *sep* is the position of the last activity that is separated from the current activity.

4 Experiments and Evaluation Results

The evaluation of the proposed methods is described in this section. The purpose of the experiments is to evaluate the effectiveness of the proposed methods and make a comparison with the results of Hidden Markov Model (HMM). The testing of activity recognition was done on two datasets, WSU dataset [5] and Kasteren dataset [2].

WSU dataset. The WSU dataset recorded the sensor data streams that 24 volunteers are assigned to perform five activities in the smart environment where 41 sensors were installed. The data are collected for 13 days with the multiple digital sensors.

```
Algorithm Activity_Recognition
Input: s[], MPL, Delay
Output: the activity label of each s[]
{   sep = 0; pos = 1; un_pos = pos;
    d = Delay – 1;
    for ( ; ; )
        $c_k$ = Disciminant_pattern(s[], sep, pos);
        while ($c_k$ == 0 and d > 0)
            pos = pos + 1;
            $c_k$ = Disciminant_pattern(s[], sep, pos);
            d = d – 1;
        endwhile
        if ($c_k$ • 0)            /*   Stage 1   */
            Label($c_k$, un_pos, pos);
            d = Delay – 1;
            if ($c_k$ • Activity of Seq[un_pos-1])
                sep = un_pos – 1;
              endif
              un_pos = pos + 1;
        else                      /*   Stage 2   */
            $c_k$ = Find_RepPattern(s[], un_pos);
            if ($c_k$ == 0)    /*   Stage 3   */
               A = Find_candidate(s[un_pos-1], s[un_pos], Activity k of s[un_pos-1]);
               $c_k$ = Frequent_Pattern(s[], sep, un_pos, A);
            endif
            Label($c_k$, un_pos, un_pos);
            if ($c_k$ • Activity of Seq[un_pos-1])
               sep = un_pos – 1;
            endif
            un_pos = un_pos + 1;
        endif
        pos = pos + 1;
    endfor
}
```

Fig. 3. The activity classification algorithm.

The activities include making a phone call, washing hands, cooking, eating, cleaning, and others. The detailed information is shown in Table 1.

Kasteren dataset. The Kasteren's dataset contains the data streams that a 26-year-old man living alone in a three-room apartment where 14 state-change sensors were installed. The data are collected for 28 days with the multiple sensors. The activities include leaving, toileting, showering, sleeping, breakfast, dinner, and others.

The detailed information of activities is shown in Table 2, the sensor readings are set to get sampling per 60 s. The time slice duration is long enough to discriminative and short enough to provide high accuracy labeling results. For give a fair evaluation on such a dataset, the 28-days dataset are separated into training set and test set using

Table 1. The activities in WSU dataset.

Activities	# of instances	% of time
Others		97.47 %
Make a phone call	24	0.36 %
Wash hands	24	0.15 %
Cook	24	0.94 %
Eat	24	0.34 %
Clean	24	0.74 %

Table 2. The activities in Kasteren's dataset.

Activities	# of instances	% of time
Others		11.50 %
Leaving	34	56.40 %
Toileting	114	1.00 %
Showering	23	0.70 %
Sleeping	24	29.00 %
Breakfast	20	0.30 %
Dinner	10	0.90 %
Drink	20	0.20 %

n-fold cross validation, which is that one full day is used to test and the other remaining days are used to train.

A daily life dataset generally contains various activities. However, because the lengths of different activities have large disparity, the evaluation of the effectiveness should concern about the ratios both of the correcting prediction in time slice and activity. The experiments are evaluated by two measures: time slice accuracy and class accuracy. The time slice accuracy stands for the percentage of correctly classified streaming data of daily time slice. The class accuracy represents the average percentage of correctly classified time slices of each activity. The two measures are defined as follows:

$$timeslice\_accuracy = \frac{\sum_{i=1}^{N} (predict(i) = true(i))}{N}, \tag{8}$$

$$class\_accuracy = \frac{1}{K}\sum_{k=1}^{K} \frac{\sum_{i=1}^{N_k} (predict(i) = true(i))}{N_k}, \tag{9}$$

where ($predict(i) = true(i)$) is a binary indicator given 1 when the predicting activity for streaming data s_i at time i is the same to the ground truth; otherwise, the value 0 is given. N is the total number of time slices. K is the number of activities. N_k is the total number of time slices with activity k.

The base-line experiments used HMM method to evaluate WSU dataset and Kasteren dataset (KD) in off-line actions using k-cross validation. For the two data sets, one day is kept for testing and the other days are the training data. The results are shown in Table 3. Generally, the time slice accuracy is better than class accuracy. The WSU dataset has high recognition rates, whereas the Kasteren dataset is not.

Table 3. The base-line results in HMM with off-line training.

Data sets	HMM	
	Time slice accuracy	Class accuracy
WSU	99.89 %	95.39 %
KD	54.89 %	45.46 %

The incremental training then is tested on the same two data sets. The data for some day are also selected from the datasets for training. However, training process was done by incrementing data day by day. This process is also completed in cross-validation. The experiments are tested on different maximal prefix length (MPL) from 1 to 4. The methods of comparison include HMM vs. μ_k^{Ave} and HMM vs. μ_k^{Max}.

The experimental results of WSU dataset are shown in Figs. 4, 5, 6 and 7. As the case of off-line training, HMM has higher accuracy than μ_k^{Ave} and μ_k^{Max}. Due to WSU dataset contains too many 'other' activities, it may dominate the classification results of HMM method. For HMM, incremental learning has no big difference from off-line learning in time slice accuracy. However, the class accuracy is lower. In WSU data set, as shown in Figs. 4 and 6, the activity estimation function μ_k^{Max} has better results than μ_k^{Ave} in time slice accuracy. The improvement of μ_k^{Max} is not obvious in class accuracy.

In Kasteren's dataset, on the contrary, the results of the activity estimation function μ_k^{Ave} are better than μ_k^{Max} in both time slice accuracy and class accuracy, as shown in Figs. 8, 9, 10, and 11. Also, both of the two activity estimation functions have higher recognition rates than HMM in time slice accuracy and class accuracy after collecting five days training data.

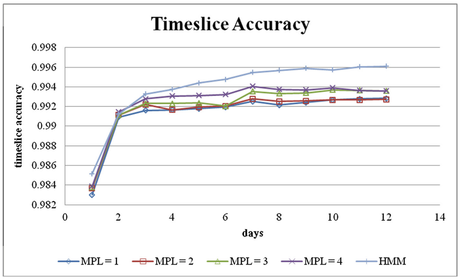


Fig. 4. WSU, time slice accuracy of μ_k^{Ave}.

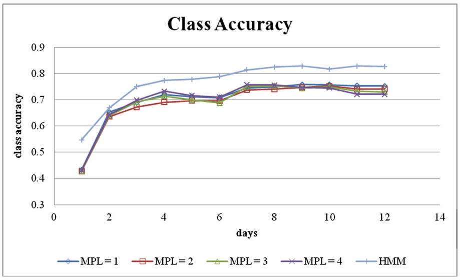


Fig. 5. WSU, class accuracy of μ_k^{Ave}.

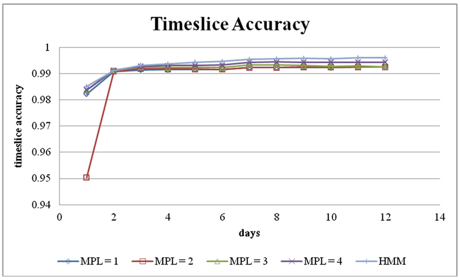


Fig. 6. WSU, time slice accuracy of μ_k^{Max}.

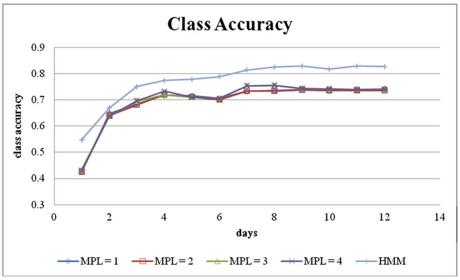


Fig. 7. WSU, class accuracy of μ_k^{Max}.

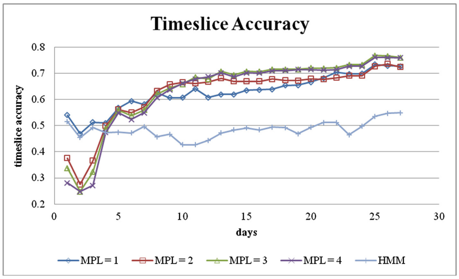


Fig. 8. KD, time slice accuracy of μ_k^{Ave}.

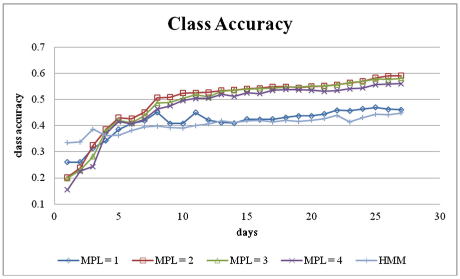


Fig. 9. KD, class accuracy of μ_k^{Ave}.

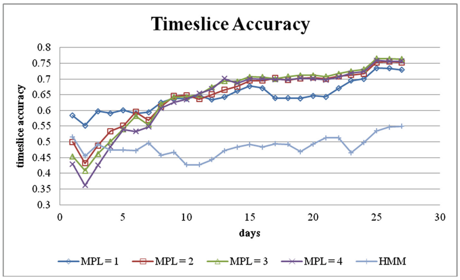


Fig. 10. KD, time slice accuracy of μ_k^{Max}.

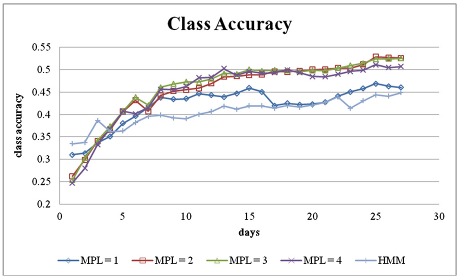


Fig. 11. KD, class accuracy of μ_k^{Max}.

For the setting of maximal prefix length (MPL), the longer MPLs have higher time slice accuracy while the training data is increasing generally. However, the better class accuracy happens as MPL = 2 while the dataset being almost learned. It is also worth to mention that there is stable accuracy when MPL = 1 although the class accuracy is not effective so much.

5 Conclusion

Smart environment is an important issue in the research area of pervasive computing. In this paper, we propose a novel activity recognition scheme based on discriminant sequence patterns and activity transition patterns for on-line detecting and incremental learning in a smart environment. First, a compressed reverse suffix tree is used to mine the discriminant sequence patterns from multi-sensor streaming data. The activity transition patterns are then analyzed to generate candidate activities. Two activity estimation functions and the activity classification algorithm are proposed to resolve the problem of activity recognition efficiently and effectively. The high accurate activity recognition is the basis of providing high-quality services for users in smart environments.

The future work is to extend the proposed scheme to handle the problem of concept drifting in multi-users' smart space and unstable environments.

Acknowledgment. This research was supported in part by the National Science Council of Taiwan, R. O. C. under contracts NSC100-2221-E-024-021 and NSC101-2221-E-024-026.

References

1. Hsu, K.C., Chiang, Y.T., Lin, G.Y., Lu, C.H., Hsu, J.Y.J., Fu, L.C.: Strategies for inference mechanism of conditional random fields for multiple-resident activity recognition in a smart home. In: The 23th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems, pp. 417–426 (2010)
2. Kasteren, T.V., Noulas, A., Englebienne, G., Kröse, B.: Accurate activity recognition in a home setting. In: The 10th International Conference on Ubiquitous Computing, pp. 1–9 (2008)
3. Singla, G., Cook, D.J., Schmitter-Edgecombe, M.: Recognizing independent and joint activities among multiple residents in smart environments. J. Ambient Intell. Humaniz. Comput. **1**(1), 57–63 (2010)
4. Yakhnenko, O., Sillvescu, A., Honavar, V.: Discriminatively trained markov model for sequence classification. In: The 5th IEEE International Conference on Data Mining, Houston, pp. 498–505 (2005)
5. WSU Datasets. http://ailab.wsu.edu/casas/datasets/index.html
6. Xing, Z., Pei, J., Keogh, E.: A brief survey on sequence classification. ACM SIGKDD Explor. Newsl. **12**(1), 40–48 (2010)
7. Lewis, D.D.: Naïve Bayes at forty: the independence assumption in information retrieval. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 4–15. Springer, Heidelberg (1998)
8. Chuzhanova, N.A., Jones, A.J., Margetts, S.: Feature selection for genetic sequence classification. Bioinformatics **14**(2), 139–143 (1998)
9. Lesh, N., Zaki, M.J., Ogihara, M.: Mining features for sequence classification. In: The 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 342–346 (1999)
10. Littlestone, N.: Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. Mach. Learn. **2**(8), 285–318 (1988)

11. Huang, R.S., Chien, B.C.: Activity recognition on multi-sensor data streams using distinguishing sequential patterns. In: The 27th Annual Conference of the Japanese Society for Artificial Intelligence, 2A1-IOS-3b-1 (2013)
12. Ji, X., Bailey, J., Dong, G.: Mining minimal distinguishing subsequence patterns with gap constraints. In: The 5th IEEE International Conference on Data Mining, pp. 194–201 (2005)
13. Keogh, E., Pazzani, M.J.: Scaling up dynamic time warping for data mining applications. In: The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 285–289 (2000)
14. Kaján, L., Kertést-Farkas, A., Franklin, D., Ivanova, N., Kocsor, A., Pongor, S.: Application of a simple likelihood ratio approximant to protein sequence classification. Bioinformatics **22**(23), 2865–2869 (2006)
15. Lin, Y.X., Chien, B.C.: A discriminant based document analysis for text classification. In: The 2010 International Computer Symposium, Workshop of Artificial Intelligence, Knowledge Discovery, and Fuzzy Systems, pp. 594–599 (2010)

Scalable Data Analytics: Theory and Applications

Using Knowledge Graph to Handle Label Imperfection

Yi Liu[1](✉), Huakang Li[2], and Yizheng Chen[1]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University,
Shanghai, China
ly0406@sjtu.edu.cn, c.yizheng@hotmail.com

[2] Department of Computer Sincere and Technology,
School of Computer Science and Technology, School of Software,
Nanjing University of Posts and Telecommunications, Nanjing, China
huakang.lee@gmail.com

Abstract. The performance of classification tasks extremely relies on data quality, while in real world label noises inevitably exists because of data entry errors, transmit errors and subjectivity of taggers. Different methods have been proposed to deal with label imperfection, including robust algorithms by avoid overfitting, filtering mechanism to remove noises and correction mechanism to revise noises. In this paper, we propose an approach based on knowledge graph to perceive and correct the label errors in training data. Experiments on a medical Q&A data set reveal that our knowledge graph based approach can be effective on promoting classification performance and data quality. The results as well show our approach can work in a relatively high noise level and be applied in other data mining tasks demanding deep understanding.

Keywords: Data quality · Knowledge graph · Label error · Classification

1 Introduction

Researches on machine learning focus on developing learning algorithms with least learning bias, thus the data quality has become the crucial issue when given a certain machine learning algorithm. Unfortunately, the real world data inevitably contains label noises (i.e. label errors) which can reduce the performance of classification in multiple aspects such as the accuracy of classifier, the time needed to train the classification model and the size of classifier. It proves that classification accuracies decline almost linearly with the increase of noise level [1].

Most label errors in training data comes from factors such as data entry errors, transmit errors and subjectivity of taggers. Many learning algorithms have mechanisms dealing with label noises. For example, pruning in decision tree algorithm can avoid overfitting caused by noises [2]. Still, when noise level is high, learning algorithms are not able to effectively deal with noises.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 345–356, 2014.
DOI: 10.1007/978-3-319-13186-3_32

Other methods try to handle the noises in data before using the data in classification, including filtering noises and correcting noises.

This paper proposed an approach based on knowledge graph technique to perceive and correct label errors in big data age. Knowledge graph is a concept proposed by Google[1] to serve for its search engine and other applications, whose core idea is utilizing ontology to simulate entities and relationships in real world to help machine understand the world intelligently [3]. Therefore we adopt this concept in noise correction to better perceive the nature of noises. We use big social data collected from medical Q&A system to validate our approach for tackling label imperfection. A study reports 83 % of internet users in the U.S. seek health information online [4] and health care systems are playing a much more essential role in recent life [5].

Our approach implements the knowledge graph on a label correction method raised by Teng et al. [6]. Concretely, naive bayes classifiers are used to recognize and modify the error labels of training data. After modifying the labels, the noise level proves to decline dramatically than before. Then we use the modified data to construct classifier for classification rather than correction, and it proves that accuracy has improved than that built from corrupt data. The main contributions of this paper are outlined as follows:

- We build a knowledge graph base containing medical entities such as diseases entities, symptom entities, medicine entities and their relationships from large scale of Q&A healthcare data, using several knowledge extraction techniques.
- We validate the effect of knowledge graph in tackling label imperfection problem by comparing our approach to other approaches. We verify our approach is more effective than other approaches on improving classification quality and data quality.
- Our approach can work at a relatively high noise level and still achieve satisfying performance.

This paper is organized as follows. Section 2 reviews the most related works in respects of label errors handling. Section 3 presents our approach to construct knowledge graph base. Section 4 describes polishing and our knowledge graph based combined approach. Section 5 describes the experiment performance and measures the affection of depth of knowledge as well. Finally, we conclude and discuss possible directions of future works in Sect. 6.

2 Related Work

Over the course of the past 20 years, solving the problem of noises in data has been the focus of much attention in the field of machine learning and data mining. Most of learning algorithms have mechanisms to diminish the impact that noises bring to the classification performance. Pruning in decision tree is used to avoid overfitting caused by noise. Wilson et al. [7,8] applied several instance-pruning

[1] http://www.google.com/insidesearch/features/search/knowledge.html

techniques which can remove noises from the training set and reduce the storage consumption. However, the performance of these learning algorithms become very bad when the noise level is very high, and classification accuracies decline almost linearly with the rise of noise level [1].

As long as the noise exists in training data, the classification quality will be affected severely. Thus, some approaches use filtering mechanisms to identify and filter the noise examples before feeding them to the classifier. Wilson et al. [9] attempted to filter the noise examples by using a 3-NN classifier as filter and apply 1-NN classifier on the filtered data as classifier. Aha et al. [10] proposed IB3 (a version of instance-based learning algorithm) to remove noise with lower updating costs and lower storage requirements. Brodley et al. [11,12] used a set of learning algorithms to construct classifiers as filters to a dataset before feeding it to classification.

However, filtering noises enhances data quality at the cost of decreasing the amount of data retained for training, and it seems pity and inappropriate to discard error label data especially when the training data is difficult to re-collect such as historical data [13]. Correcting the error labels instead of simply filtering them is a better approach that accomplishes both data quality and data amount. Zeng et al. [6] proposed a method called ADE (automatic data enhancement) which can correct label errors through numbers of iterations using multi-layer neural networks trained by using backpropagation as the basic framework. Teng et al. [13,14] introduced a noise correction mechanism called polishing and correct noises both in classes and attributes. Teng also compared polishing with filtering and traditional approach of avoiding overfitting, and proved noise correction recovers information not available with the other two approaches [14,15].

The approaches discussed above have the following limitations: (i) Some use filtering which may decrease the bulk of data. (ii) Most of these approaches do not work well at a high noise level. (iii) Most of these works only measured the promotion that their approaches bring to classification performance, yet haven't measured the exact values of data quality promotion. Therefore, we propose an approach based on knowledge graph to tackle these limitations.

3 Knowledge Graph Building

3.1 Data Source

We use big data collected from a Chinese medical social Q&A website[2] and Chinese encyclopedia website Baidu Encyclopedia (BE)[3] to build a medical knowledge base. Figure 1 shows a glimpse of a few entities and relationships in the graph.

The edge between a disease entity and a symptom entity implies that the disease have such symptom. For example, *gastritis* has *diarrhea* and *vomit* symptoms, and *fatigue* can be due to *anemia* or *Parkinson*. There are three types

[2] http://www.120ask.com

[3] http://baike.baidu.com

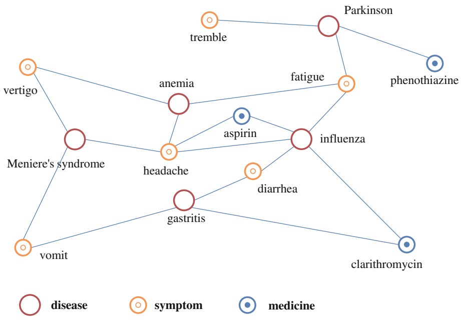


Fig. 1. A sketch map of our medical knowledge graph

of entities in knowledge graph, and two entities of the same type cannot be connected directly. This assumption is justifiable. Because in real world, two diseases are related since they share several common symptoms, two medicines are related since they can both used to treat some kind of disease. Their relationship is connected by other entities, not themselves directly.

3.2 Entities Extraction

To build the knowledge graph base, we need to extract disease entities, symptom entities and medicine entities. These are done by following steps below:

- In the first phase, we use web crawling technique to acquire disease entities, medicine entities from BE. As BE pages are well structured and tagged, we adopt Maximum Entropy classifier to classify these entities to big categories. After sort out these entities and their categories, we acquire a known entities set.
- In the second phase, we conclude linguistic patterns of entities and use these patterns to find more entities in the Q&A archives. Bootstrapping on syntactic patterns are frequently used to extract knowledge [3]. Chinese words are composed of characters, and affixes (prefixes and suffixes, contains one or more characters) usually have specific meaning about the type of words. So we use prefixes and suffixes concluded from the known entities set to find more and more entities. After acquiring these new entities, we conduct artificial selection to discard entities which do not belong to medical domain. Hence, we get a bigger set of entities than first phase.
- Then we perform several iteration of second phase and finally get a set of nearly 30,000 disease entities and 30,000 medicine entities.

Since most patients describe their symptoms orally and informally, symptoms cannot be extracted from encyclopedia website. We firstly use tfidf [16] and IG (information gain) technique [17] to find words and phrases that are more informative in the Q&A archives, and artificially select some symptom entities. Then we use bootstrapping to find more and more symptom entities. Finally we get a set of nearly 4,000 symptom entities.

3.3 Relationship Extraction

In most of the existed knowledge bases such as Wikipedia[4], the relationships between entities or relationships between entities and their attributes are established manually by experts in related field. Our knowledge base contains a relatively big amount of entities and we don't have professional knowledge in medical taxonomy. Therefore we adopt a method to automatically extract relationships between entities from big data, whose details will be discussed in Sect. 4.2.

4 Mislabel Correction

As we mentioned above, polishing proposed by Teng et al. [13,14] proves to be quite well in mislabel correction. The core idea of our approach is to adopt polishing as basic method and use information from the established knowledge graph to adjust the weight of entity features in label correction phase. Since knowledge graph maps relationships of entity features, it can be used to strengthen the more informative entity features and weaken the less informative entity features. We assume that the entity with more connection to other entities and greater co-occurrence rates with others plays a more important role in mislabel correction. Thus, they should be endowed with more weight.

4.1 Polishing

The basic polishing algorithm comprises two phases: prediction and adjustment [14]. The prediction phase aims at finding candidate training examples that are suspectable to contain error labels, while the adjustment phase decides the final changes into the candidates. The polishing algorithm can predict and correct both attributes errors and label errors (i.e. class errors), in this paper we use it to correct label errors, so we only outline this part.

In the prediction phase, a chosen learning algorithm performs K-fold cross validation. Teng et al. set K to be 10. The K-fold cross validation divides all the examples in K groups called folds, and constructs K classifiers each using K-1 folds as training set and the fold left out as test set. If the K-fold cross validation algorithm predicts a label inconsistent with the original label, this example is added to suspected candidates.

In the adjustment phase, for each example in candidates set, K classifiers constructed in the prediction phase are used to predict labels of this example. If the predicted labels of K classifiers are identical and different from the original label, polishing judges the new label to be the right one and modifies the example using the new label.

4.2 Knowledge Graph

We define our knowledge graph to be a set of vertices $(v_1, v_2, \ldots, v_m)$ and edges $(e_1, e_2, \ldots, e_m)$. Each vertex represents an entity and each edge represents direct

[4] http://www.wikipedia.org

relationship between two entities. Direct relationship means strong connection between two entity vertices. For instance, a brief example of relationships of several entities have been shown in Fig. 1, *gastritis* has symptoms of *vomit* and *diarrhea*, so they are directly connected. And the relationship between *Meniere's syndrome* and *gastritis* can not be described, we only know they share some common symptoms, so their relationship is indirect.

We define *distance* as the shortest path length between two vertices. *distance* between any two vertices can be computed once the *length* of all edges are known. The *length* of edge are computed using formula:

$$length(v_i, v_j) = \frac{1}{co\text{-}occurrence\ rate(v_i, v_j)} \tag{1}$$

co-occurrence rate can measure closeness of two entity vertices if they have direct relationship. The smaller *length* is, the larger *co-occurrence rate* is. The *co-occurrence rate* is computed from the Q&A data according to formula:

$$co\text{-}occurrence\ rate(v_i, v_j) = \frac{2 * n_{ij}}{n_i + n_j} \tag{2}$$

Here v_i, v_j represent any two entity vertices. n_{ij} represents the num of Q&A pairs in which v_i and v_j occur simultaneously, n_i defines the num of pairs in which v_i occurs, and n_j defines the num of pairs in which v_j occurs. Apparently the *co-occurrence rate* is maximum value 1 if two entities always occur simultaneously in Q&A pairs. If *co-occurrence rate* is below a threshold M, we assume the two entity vertices has no direct relationship, thus no edge existing between them.

Also, we define *related degree* to measure relationship closeness between two vertices even when they are not directly connected in the knowledge graph (namely no edge between them).

$$related\ degree(v_i, v_j) = \frac{1}{distance(v_i, v_j)} \tag{3}$$

Obviously *related degree* is equivalent to *co-occurrence rate* when there is an edge directly connecting two entity vertices. *distance* is computed using Dijkstra Shortest Path algorithm.

One advantage of knowledge graph is that we can extend or modify the graph once we grasp new knowledge through science researches. When we discover a new disease, we add it into graph and connect it to other symptoms or medicines based on the information we know about it. And if latest medical research shows some kind of medicine can help treat a disease, which hasn't be applied before, we can connect them and endow them some kind of relationship.

4.3 Weight Adjustment

We compute the weights of entity features according to formula:

$$weight(v_i) = initial\ weight + \alpha \sum_{v_j \in V, v_j \neq v_i} related\ degree(v_i, v_j), \quad \forall step(v_i, v_j) < MAXDEPTH \tag{4}$$

V is the vertices set in the graph and $MAXDEPTH$ defines the depth of relationships we mine. We define *initial weight* to be 1, and α is the adjustment factor to control the impact of knowledge graph to feature weights. The bigger α is, the greater weight adjustment will be made, and the more impact knowledge graph will have on the final weight. $MAXDEPTH$ sets a limit to which vertices to be considered when computing the weight of a vertex, namely the analysis depth of knowledge graph. We believe the weight is more precise if the depth goes deeper. However, there is a tradeoff between analysis depth and computational complexity because the related vertices number is quite large when we analysis graph quite deeply. We will conduct experiments about the affection of knowledge depth on correction labels in Sect. 4.2.

4.4 Combined Algorithm

Our approach combines polishing and weight adjustment by knowledge graph to correct noise labels in training examples. We use multinomial naive bayes (MNB) classifier as the basic classifier in K-fold cross validation. We choose MNB because it proves to be both efficient and accurate for text classification tasks [18]. Still, MNB makes a poor assumption that features of examples are independent of others, which is clearly unreasonable in most real-world tasks. We adjust feature weights in MNB classifier according to knowledge graph to make up this assumption. Weights of entity features are calculated according to formula (4) and weights of other features are defined as 1. When the corrupt training data is prepared, we adjust the weight of features in the training examples, and get the adjusted training data. Then we utilize this data to following the same procedures of polishing in Sect. 4.1. We also set K to be 10 in the K-fold cross validation. Afterwards we can obtained data corrected by our combined approach.

5 Experiment Result

5.1 Data Sets

As we mentioned above, our data is extracted from a huge set of nearly 20 million medical Q&A pairs. The format of data is specified in Table 1, each example has a description text which patients depicts about their circumstances and symptoms, and each example has a department label showing the department where this patient should be treated. The description text of Q&A pairs is usually short, less than 200 characters. The whole data sets contain more than 10 departments, including ‘obstetrics and gynaecology’, ‘internal medicine’, ‘surgery’, ‘pediatrics’, ‘dermatology’, ‘ophthalmology and ‘otorhinolaryngology’,

Table 1. The format of Q&A pairs

Description	Answer	Department
I play badminton and when I use backhand serve, my hand tremble. My brachioradialis hurts too...	It may be caused by overexercise, I suggest you see a bone surgery doctor to...	Surgery

'psychology', 'traditional Chinese Medicine', 'infectious diseases', 'oncology' and 'plastic surgery'. We use our approach to perceive and correct the error department labels in training examples. Since the corpus is in Chinese, we use several NLP methods specialized in handling Chinese text: tokenizing Chinese text and transfer traditional Chinese characters to Chinese simplified characters. Afterwards, we extracted approximately 200,000 features from the raw data. Finally, we get nearly 9,725,000 training examples. In order to get the corrupt data, we artificially corrupt the data with random label noises.

5.2 Evaluation Measures

We choose accuracy as the measure metric to evaluate the classification quality improvement after label correction. And we use several metrics to evaluate the data quality promotion. These metrics are *noise reduction rate* (NRR), *precision* and *recall*. As our approach and polishing correct labels by the judgement of 10 classifier voters, the changes made to the examples are not always right. So these metrics are used to evaluate these changes. NRR is defined in (5) and measures the noise level decrease after label correction. *precision* measures the percentage of right changes in the whole changes made by label correction approach. *recall* measures the percentage of error labels which is actually corrected. It's obvious that NRR most intuitively reflects the data quality promotion.

$$NRR = noise\ level\ in\ origin\ data - noise\ data\ in\ corrected\ data \tag{5}$$

We use three methods: *Unpolishing*, *Polishing* and *Polishing* + KG in classification accuracy comparison. *Unpolishing* approach uses the unmodified corrupt data to build classifier. *Polishing* approach uses the data corrected by polishing method to build classifier. And *Polishing*+KG approach uses the data corrected by our approach to build classifier. All the three approaches are applied in accuracy comparison, and the latter two are applied in mislabel reduction rate comparison. In addition, we set $MAXDEPTH$ to 1 in *Polishing* + KG when compared with other two approaches.

5.3 Classification Accuracy

We compare the classification accuracy on training data produced by three approaches mentioned above. For each approach, 10-fold cross validation is performed on data to obtain classification accuracy. In each trial, nine folds are used

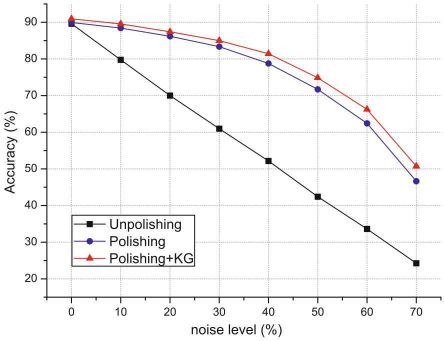


Fig. 2. A comparison accuracy on data by $Unpolishing$, $Polishing$ and $Polishing + KG$ on the medical Q&A data set

for training data to test the accuracy of the rest fold. The final accuracy is the average accuracy of 10 trials. Here we use cross validation to evaluate classification accuracy, different from label correction phase where cross validation is used to pick up candidates and construct classifiers as voters. We choose cross validation to validate accuracy because it can reduce the risk of overfitting on the test set.

Figure 2 shows the comparison of three approach on classification accuracy at different noise levels. For $Unpolishing$ approach, accuracy declines almost linearly with noise level increase. At most cases, the improvement of $Polishing$ and $Polishing + KG$ on $Unpolishing$ is quite significant, the performance of $Polishing$ is 10 %–30 % higher than $Unpolishing$, while our approach $Polishing + KG$ acquires accuracy usually 1 %–4 % higher than the pure $Polishing$. We can see noise data cut down accuracy dramatically when no correction is conducted. $Polishing$ corrects part of the error labels and provides a much higher accuracy. Furthermore, $Polishing + KG$ approach mines the relationships between entity features and endows more weights to the more informative ones, so it achieves better accuracy score than $Polishing$. Particularly, at noise level of 0 %, the improvements of $Polishing$ and $Polishing + KG$ are both not remarkable, $Polishing$ is merely 0.3 % higher than $Unpolishing$, and $Polish + KG$ is 1.3 % higher than $Unpolishing$, we believe $Polishing + KG$ also has effect on improving classification accuracy even when data is nearly noise-free.

5.4 Data Quality Promotion

We compare the classification-independent metrics to test data quality promotion by $Polishing$ and $Polishing + KG$ approach. When we artificially corrupt the data, we have made mark to every example what is the true label of it. After label correction by two approaches, we test the precision, recall and NRR according to these marks. We use NRR as the main metric on data quality promotion, while the other two help us to understand and explain the promotion.

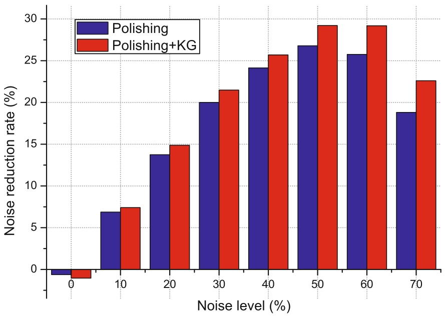


Fig. 3. A comparison of noise reduction rate (NRR) by *Polishing* and *Polishing* + *KG* on the medical Q&A data set

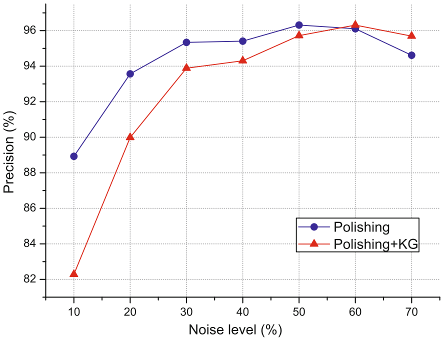


Fig. 4. A comparison of precision by *Polishing* and *Polishing*+*KG* on the medical Q&A data set

Figure 3 shows NRR by two approaches. NRR of *Polishing* + *KG* is approximately 1 %–4 % higher than *Polishing*. It seems odd that NRR is negative at noise level of 0 %, which means the noises increase after label correction. However, this phenomenon can be explained. At noise level of 0 %, we assume data to be noise-free, while data can't be completely noise-free in real-world. So it seems reasonable that *Polishing* and *Polishing* + *KG* has modified some labels which are quite possibly error labels. Generally speaking, it is shown that *Polishing* has great significance in data quality promotion and *Polishing* + *KG* achieves better performance on the basis of *Polishing*.

Figures 4 and 5 show the precision and recall. We do not considerate precision and recall at noise level of 0 % because it's meaningless. At most noise levels, precision of *Polishing* + *KG* is lower than *Polishing*, however the recall of *Polishing*+*KG* is much higher than *Polishing*. Usually precision and recall have a contradiction relationship that precision decreases along when recall increases. So it's reasonable that *Polishing* + *KG* has a lower overall precision. When noise level is quite higher, the precision and recall of *Polishing* + *KG* are both higher than *Polishing*. We assume this is due to that knowledge diminishes the interference of noises, the effect is more remarkable when the noise level is higher.

5.5 Knowledge Depth Affection

We conduct experiment of how knowledge depth affects the results. According to (3), we adjust the entity weights by computing closeness of an entity to other entities. We believe the bigger $MAXDEPTH$ is, the more precise weights will be obtained. This thought is driven by that we get more information about something when we recognize it more deeply. Figure 6 shows the accuracy comparison of different knowledge depth from 1 to 3. The accuracy improves 0–1.3 % when knowledge depth grows from 1 to 2 at different noise levels, while the accuracy

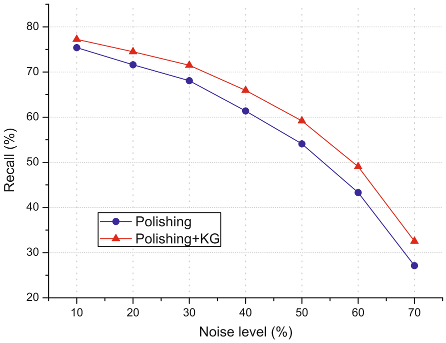


Fig. 5. A comparison of recall by *Polishing* and *Polishing+KG* on the medical Q&A data set

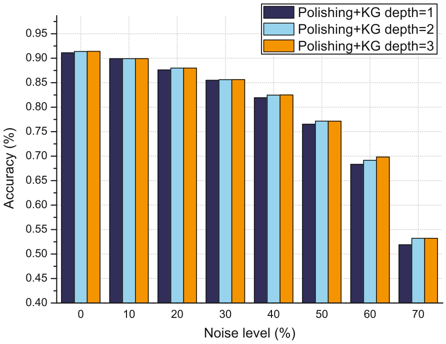


Fig. 6. knowledge depth affection on accuracy

improvement is tiny when depth grows from 2 to 3. When knowledge depth grows, the amount of relationships of one entity to others grows rapidly and more weights are endowed to the more informative ones. The results shows deep knowledge perception can enhance classification performance.

6 Conclusion

In this paper, we propose a knowledge graph based approach combined with polishing to handle label imperfection problem. This method differs from previous statistical methods in that it tries to cognize the data in a way similar to the real world. Experiment results demonstrate our approach has effect on boosting classification performance and data quality. It can effectively correct mislabels even under the circumstance of a quite high noise level to approximately 60 %. Beside handling the noise data, the knowledge graph technique we used can be applied in feature selection in classification as well.

Our future work will be focus on ameliorating the graph by establishing more types of entities and more detailed relationships in it, like establishing both positive and negative relationships between two entities. The More researches will be conducted to recognize data noises in a more human-like rather than machine-like way. In addition, we shall apply our approach to other fields such as social networks and business data analysis.

Acknowledgements. This work was supported by the NSFC (No. 61272099, 61261160502 and 61202025), Shanghai Excellent Academic Leaders Plan (No. 11XD1402900), the Program for Changjiang Scholars and Innovative Research Team in University of China (IRT1158, PCSIRT), the Scientific Innovation Act of STCSM (No. 13511504200), Singapore NRF (CREATE E2S2), and the EU FP7 CLIMBER project (No. PIRSES-GA-2012-318939).

References

1. Zhu, X., Wu, X.: Class noise vs. attribute noise: a quantitative study. Artif. Intell. Rev. **22**(3), 177–210 (2004)
2. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
3. Wu, W., Li, H., Wang, H., Zhu, K.Q.: Probase: a probabilistic taxonomy for text understanding. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 481–492. ACM (2012)
4. Zhang, Y.: Contextualizing consumer health information searching: an analysis of questions in a social Q&A community. In: Proceedings of the 1st ACM International Health Informatics Symposium, pp. 210–219. ACM (2010)
5. Kunz, H., Schaaf, T.: General and specific formalization approach for a balanced scorecard: an expert system with application in health care. Expert Syst. Appl. **38**(3), 1947–1955 (2011)
6. Zeng, X., Martinez, T.R.: An algorithm for correcting mislabeled data. Intell. Data Anal. **5**(6), 491–502 (2001)
7. Wilson, D.R., Martinez, T.R.: Instance pruning techniques. In: ICML, vol. 97, pp. 403–411 (1997)
8. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Mach. Learn. **38**(3), 257–286 (2000)
9. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. Syst. Man Cybern. **2**(3), 408–421 (1972)
10. Aha, D.W., Kibler, D.F.: Noise-tolerant instance-based learning algorithms. In: IJCAI, Citeseer, pp. 794–799 (1989)
11. Brodley, C.E., Friedl, M.A.: Identifying and eliminating mislabeled training instances. In: AAAI/IAAI, Citeseer, vol. 1, pp. 799–805 (1996)
12. Brodley, C.E., Friedl, M.A.: Identifying mislabeled training data (2011). arXiv preprint arXiv:1106.0219
13. Teng, C.M.: Evaluating noise correction. In: Mizoguchi, R., Slaney, J.K. (eds.) PRICAI 2000. LNCS, vol. 1886, pp. 188–198. Springer, Heidelberg (2000)
14. Teng, C.M.: Polishing blemishes: Issues in data correction. IEEE Intell. Syst. **19**(2), 34–39 (2004)
15. Teng, C.M.: A comparison of noise handling techniques. In: FLAIRS Conference, pp. 269–273 (2001)
16. Li, J., Zhang, K., et al.: Keyword extraction based on tf/idf for chinese news document. Wuhan Univ. J. Nat. Sci. **12**(5), 917–921 (2007)
17. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: ICML, vol. 97, pp. 412–420 (1997)
18. McCallum, A., Nigam, K., et al.: A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization, Citeseer, vol. 752, pp. 41–48 (1998)

Learning to Display in Sponsored Search

Xin Xin(✉) and Heyan Huang

School of Computer Science and Technology,
Beijing Institute of Technology, Beijing, China
{xxin,hhy63}@bit.edu.cn

Abstract. In sponsored search, it is necessary for the search engine, to decide the right number of advertisements (ads) to display for each query, in the constraint of a limited commercial load. Because over displaying ads will lead to the commercial overload problem, driving some of the users away in the long run. Despite the importance of the issue, very few literatures have discussed about how to measure the commercial load in sponsored search. Thus it is difficult for the search engine to make decisions quantitatively in practice. As a primary study, we propose to quantify the commercial load by *the average displayed ad number per query*, and then we investigate the displaying strategy to optimize the total revenue, in the constraint of a limited commercial load. We formalize this task under the framework of the secretary problem. A novel dynamic algorithm is proposed, which is extended from the state-of-the-art multiple-choice secretary algorithm. Through theoretical analysis, we proof that our algorithm is approaching the optimal value; and through empirical analysis, we demonstrate that our algorithm outperforms the fundamental static algorithm significantly. The algorithm can scale up with respect to very large datasets.

1 Introduction

The research in sponsored search has attracted more and more attention nowadays [11]. Search engines mainly depend on such kind of advertising to obtain the revenue. In the pay-per-click mechanism, the search engine agency will get paid by the advertiser, once an ad is clicked by the user. It is reported that over 1,200 million US dollars[1] have been spent in the US mobile advertising market in 2012, which is predicted to be up to 5,000 million in 2016.

One important issue in sponsored search is how to assign the right number of ads to display for each query in order to optimize the total revenue, in the constraint of a limited commercial load. Recent research [2] shows that displaying too many ads will "train" the user to ignore the ads in the result page, which will finally impair the utilities of the advertiser and the search engine agency. Some previous work also supports this point by the following evidences. Users have been reported to show bias against sponsored results after they know the

[1] www.emarketer.com

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 357–368, 2014.
DOI: 10.1007/978-3-319-13186-3_33

insight mechanism [8]; from a user study [5], when sponsored results are as relevant as organic results, more than 82 % of users will see organic results first; and organic results have also been demonstrated to obtain much higher click-through rate (CTR) than sponsored results. Therefore, if the commercial overload problem is not paid attention to, some of the users will be driven away in the long run, and the balance of the whole system will be destroyed.

Although limiting the commercial load in designing the displaying strategy is an important issue, in previous literatures, the problems of how to measure the commercial load and how to determine the displaying strategy in the constraint of a limited commercial load, have rarely been investigated sufficiently. Previous related research is mainly conducted from the field of information retrieval. Typical work includes the CTR prediction (regression) and the relevance classification. In these methods, an ad is recognized as "proper" to display, if the CTR or the relevance score is higher than a threshold. Previous work indeed provides informative statistics from the field of information retrieval; however, as sponsored search is an inter-discipline subject, in designing the displaying strategy in practice, the search engine also needs to measure the commercial load quantitatively, from the field of operational research. Without quantifying the commercial load, it is very difficult to define what is the reasonable threshold value, to identify whether displaying an ad is proper. In limiting the commercial load, the search engine agency has to know at least how much load they give to the user by the current system quantitatively. Then, optimizing the displaying strategy to obtain more revenue, in the constraint of a certain load rate, will be conducted. Unfortunately, these problems have not been solved thoroughly in the previous work.

In this paper, we propose to quantify the commercial load by *the average displayed ad number per query*, as a primary study, and then we investigate the displaying strategy to optimize the total revenue in the constraint of a limited commercial load. The work does not conflict with previous work. As we will show later, it is a complement from the field of operational research. Specifically, we formalize this task under the framework of the secretary problem. The fundamental static algorithm cannot obtain reliable performance, due to the unstable property of the data which would be detailed later. Inspired by the solution from the secretary problem community, we extend the previous multi-choice secretary algorithm [6], and propose a novel dynamic algorithm for this task. Through theoretical analysis, the expectation of the performance is approaching the optimal value. Experimental results in a real-world dataset also demonstrate that it outperforms the static algorithm significantly.

2 Problem Definition

2.1 Background

As shown in Fig. 1, in the search engine, queries arrive in a sequential stream. For each query q_i, the search engine will generate a rank of relevant ads to fit the t ad slots ($t = 3$ in the figure). The rank is generated according to the estimated

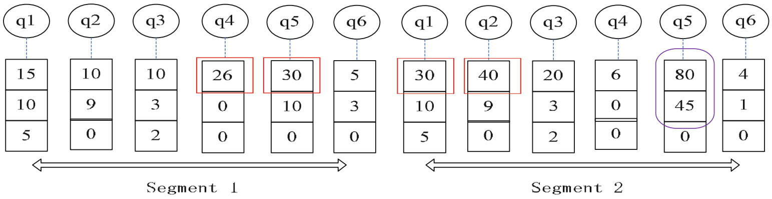


Fig. 1. An example for the learning to display problem

revenue of each query-ad pair. In general, the larger the revenue value, the higher position the ad would be placed at. Once the rank is fixed, we utilize $f(q_i, ad_j)$ to denote the estimated revenue of the j^{th} ad slot for q_i. Previous work mainly targets at constructing the function f. For example, f could be implemented by the product of the predicted CTR and the bidding price, or the relevance score calculated by a classifier. In Fig. 1, we suppose the number in each ad slot to be the estimated revenue. If there are less than t ads, the blank slot has the estimated revenue of zero.

2.2 Problem Formulation

The problems to be solved in this paper are: (1) how to measure the commercial load in sponsored search; and (2) how to assign the displayed ad number for each query to optimize the total obtained revenue, in the constraint of a limited commercial load.

Definition 1 (Query Stream Segment). *A query stream segment is a fixed number of continuous queries, together with the ads in the sequential query stream. The fixed number is also called the segment length. In Fig. 1, there are two query stream segments with the length $N = 6$.*

Definition 2 (Commercial Load Rate λ). *The commercial load rate is defined as the average displayed ad number per query. In a query stream segment with the length N and the commercial load rate λ, the total displayed ad number is equal to $N * \lambda$. For example, in Fig. 1, if $\lambda = 0.33$, only up to two ads can be displayed within each segment.*

Definition 3 (Displaying Strategy $I(q_i)$). *An displaying strategy $I(q_i)$ is to decide how many ads to display for q_i in a query stream segment, at the moment that q_i has arrived but q_{i+1} has not arrived.*

The return value of $I(q_i)$ is an integer from 0 to t, meaning displaying 0 to t ads for q_i. An important characteristic of I is, in deciding $I(q_i)$, q_{i+1} has not arrived yet. In practice, when a query arrives, the strategy should decide how many ads to display immediately without knowing the future query. This property makes the optimization problem be under the framework of the secretary problem.

Definition 4 (Learning to Display Problem). *The task of learning to display is to find the best displaying strategy I that can maximize the sum of estimated revenue within a query stream segment, in the constraint that the total displaying number is equal or less than $N * \lambda$, formulated by*

$$\max_{I} \sum_{i=1}^{N} \sum_{j=1}^{I(q_i)} f(q_i, ad_j), s.t. \sum_{i=1}^{N} I(q_i) \leq N * \lambda.$$

As shown in Fig. 1, the best solution to the problem in *Segment* 1 is,

$$I(q_1) = 0, I(q_2) = 0, I(q_3) = 0, I(q_4) = 1, I(q_5) = 1, I(q_6) = 0.$$

From the problem definition, we could see the complementary property between previous work and our work. Previous work focuses on how to accurately calculate the estimated revenue f for each ad (e.g., from the CTR multiplied by the bidding price, the relevance of the query-ad pair, the advertisability of the query, and etc.), from the field of information retrieval; but our work focuses on how to optimize the total revenue in the constraint of a limited commercial load, from the field of operational research. Both should be solved in practical scenarios.

Besides this formulation, we also considered to utilize *the average displayed ad number per user* to define this problem. The difference to our problem is that given the user number, we first need to predict how many queries will be issued by these users; and then the problem is the same as our defined problem on how to display the ads given the total ads number. Thus we try to solve this problem first and leave the prediction task as future work.

2.3 A Fundamental Static Solution

A fundamental static solution is to learn a static threshold by the information in previous segments. For example, as shown in Fig. 1, if we want to select two ads in each segment, we can utilize the second highest ad value in *Segment* 1 as the static threshold, *threshold* = 26, in determining the displaying strategy in *Segment* 2. If the value of an ad in *Segment* 2 is higher than the threshold, display it; or otherwise hide it. Execute the process until two ads are displayed.

This solution is very intuitive. If the data in the query stream is stable, this method will be quite effective. The limitation lies in that we cannot assume the data is always stable. For example, prices are going up in an irregular manner in general. Queries change with the change of seasons. The threshold in the previous segment is no longer suitable for the current one in most cases.

Take the example in Fig. 1. By using the static solution, the ads valued 30 and 40 would be selected to display in *Segment* 2. However, this is not the optimal solution. Because there is a query q_5 that has the ads valued 80 and 45. The best solution is to select these two ads. But since the static algorithm does not retain any flexibility for the change, it fails to find the best solution.

2.4 Evaluation Metric

Following evaluation metrics in the decision theory, we utilize "error ratio", which is $1 - ratio\, regret$ [7], as the evaluation metric. The error ratio is defined as

$$error\, ratio = 1 - regret = \frac{\sum_{i=1}^{N} \sum_{j=1}^{I_{best}(q_i)} f(q_i, ad_j) - \sum_{i=1}^{N} \sum_{j=1}^{I(q_i)} f(q_i, ad_j)}{\sum_{i=1}^{N} \sum_{j=1}^{I_{best}(q_i)} f(q_i, ad_j)},$$

where $\sum_{i=1}^{N} \sum_{j=1}^{I(q_i)} f(q_i, ad_j)$ is the sum of all estimated revenue selected by algorithm I, and I_{best} is the best solution. The error ratio ranges from 0.0 to 1.0. The metric is a measurement of error, thus the smaller the value, the better the performance. In the example in Fig. 1, for *Segment* 2, if $\lambda = 0.33$, the best solution is $80 + 45 = 125$ with the zero error ratio. The solution from the static algorithm is $30 + 40 = 70$, thus the error ratio is $(125 - 70)/125 = 0.44$.

3 Proposed Dynamic Algorithm

If we make the assumption that queries arrive randomly, the problem is under the framework of the secretary problem. Thus ideas from the secretary problem community can be employed. Different from the classical secretary problem in which applicants arrive one by one, in our case, a group of t ads arrive as a group. Thus, we call the problem in our case as t-tuples multi-choice secretary problem, which has never been investigated before. The most similar problem is 1-tuple multi-choice problem. Previous work [6] has proposed a recursive algorithm. The expected performance of this algorithm can achieve $(1 - O(1/\sqrt{k}))v$, where v is the optimal value and k is the number of elements to be selected. In this paper, we make an extension of this algorithm to fit the t-tuples case in our scenario, and give the theoretical and empirical analysis.

The proposed algorithm is a recursive process.

- If $k = 1$, we observe the first N/e queries without any selection. But we record the estimated revenue of each ad slot. An initial threshold is set to the largest observed value. Then for the rest $(N - N/e)$ queries, we select the first ad whose value is larger than this threshold (if all the values are less than the threshold, we select the last one).
- If $k > 1$, we sample an m from the binomial distribution $m \sim B(N, 0.5)$, and then recursively select $k/2$ ads from the first m queries in the query stream. For the rest $(N-m)$ queries, we set the threshold to the $k/2^{th}$ largest observed value in the m queries, and then select the first $k/2$ ads whose values are larger than the threshold.

We assume that in the t slots, the revenue of the lower position (close to the end position) is always less than the revenue of the higher position (close to the top position). Under this assumption, the above algorithm cannot generate irregular cases such as the second ad is selected without the first one selected.

We utilize the previous example to demonstrate the algorithm. As shown in Fig. 2, a segment of six queries arrive in a sequence. We set $\lambda = 0.33$ to select two ads from the data.

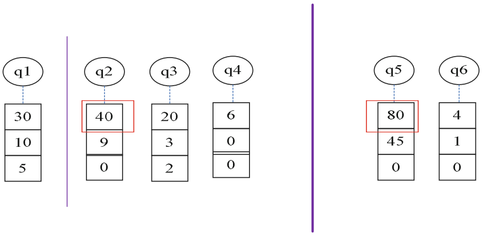


Fig. 2. An example of the proposed dynamic algorithm

- Since $k = 2 > 1$, we sample an m from $B(6, 0.5)$, and suppose $m = 4$.
- By recursively process the first four queries, we find $k = 1$. Thus we observe the first $4/e = 1$ query. We set $threshold = 30$ and select 40 in the remaining ads of the first four queries. The observed value list we maintained is "$40 > 30 > 20 > 10 > 9 > 6 > 5 > 3 > 2$".
- For the last two queries, we set the threshold to the $2/2 = 1^{th}$ value in the list, $threshold = 40$. Then we select the first one that is larger than the threshold. 80 will be selected.
- The algorithm ends, and the ads valued 40 and 80 will be selected.

From this example, we can see that the error ratio of the proposed dynamic algorithm is $(125-120)/125 = 0.04$, which is much better than the previous static method. The reason is that the proposed algorithm will change the threshold dynamically according to the observed data in the query sequence. When the observed values are lower than usual, from the maintained list, the threshold will be updated by a lower value; and when the observed values are higher than usual, the threshold will also be updated by a higher value. Therefore, when the data is not stable, this dynamic approach can fit the change in the data stream.

In the above algorithm, the time complexity is $O(N \log N)$, and the space complexity is $O(N)$, where N is the segment length. It can be utilized in large-scale data applications.

Compared with the fundamental static method, the dynamic method performs better in fitting the change of the data; however, the pure dynamic algorithm throws out all the history information, which might also impair the performance. Consequently, it is natural to combine the static algorithm and the dynamic algorithm for another improvement. In doing this, we propose to set a lower bound and an upper bound in the dynamic algorithm, based on the threshold of the static algorithm. In case that the estimated revenue is below the lower bound, it cannot be selected in any case; and similarly, in case that it is beyond the upper bound, it can be selected without comparison with the threshold. The lower bound and the upper bound are defined empirically.

4 Theoretical Analysis

In our approach, we read a permutation of t-tuples of size n as a permutation of size tn by simply expanding the tuples into t consecutive elements from the problem of 1-tuple multi-choice secretary problem. We call a permutation on t-tuples *t-monotone*, if each t-tuple $(a_1, a_2, ..., a_t)$ in the permutation satisfies $a_1 > a_2 > ... > a_t$. If we can sort each t-tuple in a permutation from the largest to smallest respectively, we call it the corresponding t-monotone permutation of the original. The differences between our algorithm and the previous one [6] are: (1) our algorithm will divide the elements at the boundary of the tuples; and (2) the assumption on the input distribution is different. We first show our algorithm works well on the old ideal distribution, and then deal with the second point.

Theorem 1. *Our algorithm has the expected competitive ratio of* $1 - O(1/\sqrt{k})$ *on the uniform distribution of all permutations of size* tn*.*

Proof. For an input permutation of size tn, the algorithm divides the input into two parts by generating a random integer m according to binomial distribution $B(n, 1/2)$. Let $Y = (y_1, y_2, \ldots, y_{tm})$ be the first part consisting of tm elements and $Z = (z_1, z_2, \ldots, z_{t(n-m)})$ be the rest. $k/2$ elements will be selected from Y in a recursive fashion. After that, the first $k/2$ elements which are larger than the $k/2$-th largest element in Y will be selected from Z. Let v be the optimal result. We follow the idea in previous work [6] by stating two lemmas.

Lemma 1. *The sum of the largest* k *elements in* Y *has expected value* $\geq (1/2 - \frac{1}{4\sqrt{k/t}})v$.

Lemma 2. *The first* $k/2$ *elements in* Z *that are larger than the* $k/2$*-th largest element in* Y *has expected value* $\geq (1/2 - 1/\sqrt{k})v$.

Since t is a very small integer const (usually less than 10) compared with k and N in practice, by using Lemma 1 and induction hypothesis, we know the expected value of the output is $\geq (1/2 - 1/4\sqrt{k/t}) \cdot (1 - O(1/\sqrt{k}))v = (1/2 - O(1/\sqrt{k}))v$. Summing up with Lemma 2, we finish the proof.

We sketch the proof for the first lemma. The proof of the second is the same as previous work [6].

Proof (of Lemma 1). Let T be the set of the k largest elements in the input. Their sum is v. We know that Y is a r.v. uniformly distributed in all the subsets whose sizes are divisible by t. Thus, conditioned on $|Y \cap T| \equiv j \pmod{t}$ for some fixed $j \in \{0, 1, 2, ..., t-1\}$, $Y \cap T$ will be a uniformly selected subset of T with size modular t equals to j. Let $B^j(k, 1/2)$ denote the distribution of $|Y \cap T|$ conditioned on j. We can observe that it is statistically dominated by $B(k, 1/2)$, implying all the moments of the former distribution is bounded by the moments of the latter. Thus $1 - o(1)$ mass of $B^j(k, 1/2)$ will be concentrated around $\sqrt{k}$ of its expectation. Also, conditioned on the event $|Y \cap T| = r$, the expected sum of $Y \cap T$ is $r/k \cdot v$. So the expected value of the $k/2$ largest elements of Y is bounded below by

$$\sum_{j=0}^{t-1} \Pr\left[|Y \cap T| \equiv j \pmod t\right] \cdot$$
$$\left(\sum_{t|r} \Pr\left[|Y \cap T| = r | r \equiv j \pmod t\right] \cdot (\min(r, k/2)/k)v\right)$$
$$\geq \sum_{j=0}^{t-1} \Pr\left[|Y \cap T| \equiv j \pmod t\right] \cdot \left(1 - \frac{1}{2\sqrt{k/t}}\right)\frac{v}{2} \quad \geq \left(1 - \frac{1}{2\sqrt{k/t}}\right)\frac{v}{2},$$

which completes the proof.

Intuitively when the input is sampled from t-monotone permutations, the output of the algorithm should be better, since the larger elements are elevated to the earlier part of the permutation. We capture this by the following lemma.

Lemma 3. *The execution of our algorithm on a corresponding t-monotone permutation always yields better output than the original permutation.*

Proof. We prove this by induction. The algorithm divide the input into two parts, selecting $k/2$ elements on the latter part while recursing into the former. By induction hypothesis, the algorithm outputs better answer on the former part. On the latter part we know the largest $k/2$ elements in the former part is the same as the original permutation. When selecting $k/2$ elements in the latter part, scanning larger elements before smaller ones will never decrease the output, implying better output. For the induction base case where $k = 1$, the classical secretary algorithm is used to select the largest element in the permutation. The algorithm will select the largest value in the original permutation but not in the t-monotone permutation iff the largest element is the n/e-th element and not the first element in a t-tuple. However this will only happen with probability less than $1/n(o(1/\sqrt{k}))$, which could be ignored.

We know that each t-monotone permutation will have exactly $(t!)^n$ corresponding permutations. Thus by mapping the original probability space into the new one, we have the following corollary.

Corollary 1. *The expected selected value over t-monotone permutations has competitive ratio* $1 - O(1/\sqrt{k})$ *on input of t-monotone permutations.*

This means that when k goes to infinity, our algorithm could obtain the optimal answer with probability 1.

5 Empirical Analysis

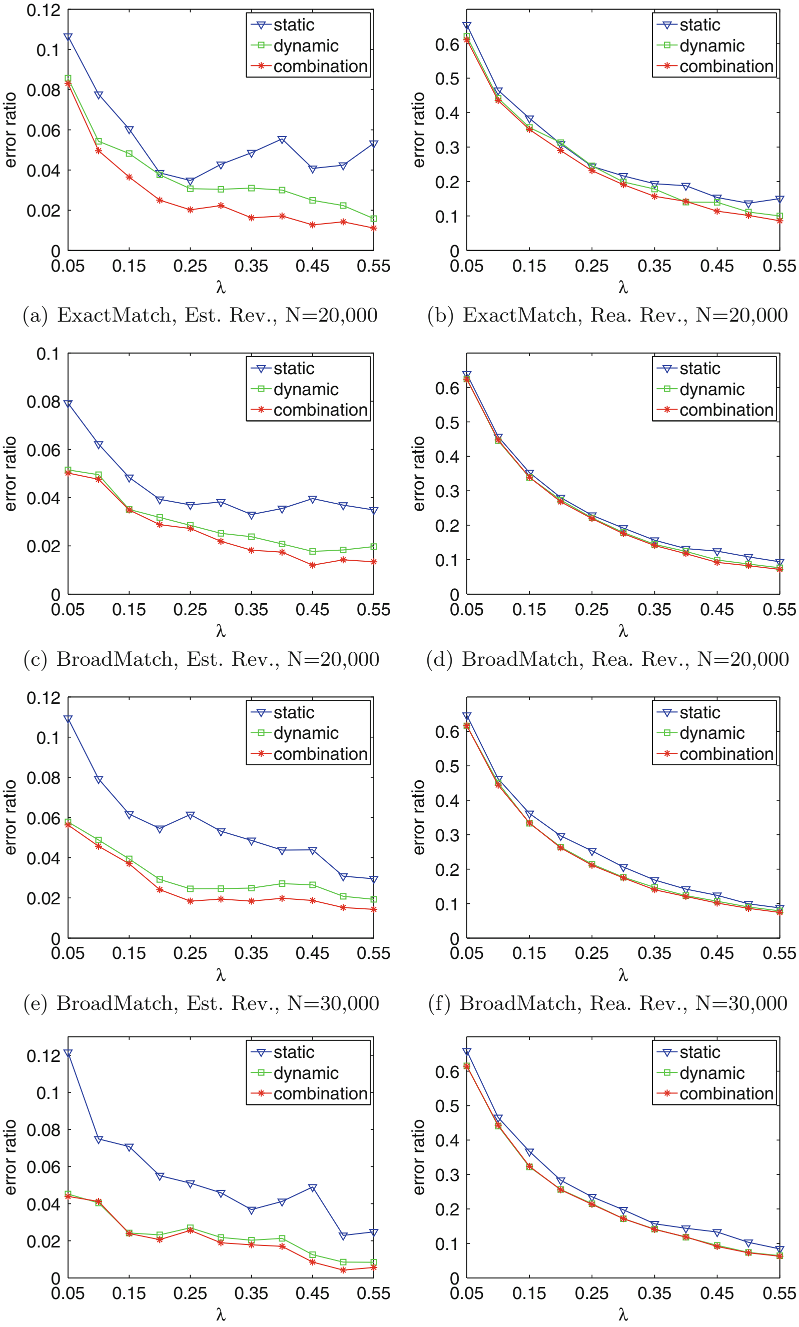
The theoretical analysis is based on the assumption that the revenue of each ad slot is randomly ordered. This assumption on the input is mathematically nice, but a more important question is whether this assumption is true in reality. So in this section, we analysis this problem with the help of the real data.

Our dataset is the click-through log from a search engine. It contains over 10 million sessions in total. In each session, there is a query and the displayed ads. There are three ad slots for each query in the system. The data is collected following the time sequence. More statistics of the dataset are summarized in Table 1. The dataset is divided into exact-match (ExactMatch) data and broad-match (BroadMatch) data. In the former, the query exactly matches the bidding keywords of the displayed ads; but in the latter, the query is only relevant but not matched with the bidding keywords. We set the segment length $N = 20,000$ without loss of generality with both datasets. Moreover, we also show the experimental results when we set $N = 30,000$ and $N = 40,000$ with the broad-match data to verify robustness of our algorithm. Similar results will be obtained on the exact-match data. Experiments are conducted on different configurations on λ from 0.05 to 0.55 with the interval of 0.05. The average error ratio is calculated among all the segments. For ad revenue estimation function f, in this paper, it is calculated by the predicted CTR multiplied by the bidding price. We employ the click chain model [4] for CTR prediction. In this model, it carefully considers the position bias in the CTR statistics. For more details, please refer the original paper.

We compare the performances among the static algorithm, the proposed dynamic algorithm, and the combination algorithm. Figures 3(a), (c), (e) and (g) shows the experimental results. It is observed that as λ becomes larger, all the algorithms perform better. This is expected, because selecting more elements is easier than selecting only a few elements. For an extreme example, if we select 10 ads from 10 ads, all strategies will get the ideal performance. Thus as λ becomes larger, the differences of strategies are getting smaller. This is why we ignore λ when it is larger than 0.55. From Figs. 3(a) and (c), it can be concluded that the dynamic algorithm consistently outperforms the static method significantly in both datasets. The improvement is 31.6 % in average with the exact-match dataset, and 33.5 % with the broad-match dataset. In addition, the combination algorithm consistently outperforms the dynamic algorithm by

Table 1. Statistics of the dataset

Query freq.	# Unique query	# Session	Avg CTR
1	1695146	1695146	0.0212
2	1058697	1342854	0.0153
3–4	772810	1275067	0.0141
5–8	262555	925065	0.0151
9–17	128304	880444	0.0162
18–32	47981	685582	0.0179
33–221480	48582	4427998	0.0201



(a) ExactMatch, Est. Rev., N=20,000

(b) ExactMatch, Rea. Rev., N=20,000

(c) BroadMatch, Est. Rev., N=20,000

(d) BroadMatch, Rea. Rev., N=20,000

(e) BroadMatch, Est. Rev., N=30,000

(f) BroadMatch, Rea. Rev., N=30,000

(g) BroadMatch, Est. Rev., N=40,000

(h) BroadMatch, Rea. Rev., N=40,000

Fig. 3. Overall performance

another significant improvement. The improvement is 25.87 % in average with the exact-match dataset, and 11.2 % with the broad-match dataset. This demonstrated that the dynamic algorithm is effective in optimizing the revenue. From Figs. 3(c), (e) and (g), it can be observed that when N becomes larger, the static algorithm does not perform better constantly. This verified the unstable property of the real data. The data changes in nature, and the old threshold might not fit the new data. Nevertheless, the proposed dynamic algorithm is performing better constantly. As shown in the theoretical analysis, when N becomes larger, k also becomes larger. This will reduce the error theoretically. This demonstrated that the proposed algorithm is robust in different configurations.

It might be argued that all the algorithms depend on the revenue estimation function f. In practice, the function usually have large errors. For example, it is very hard to accurately estimate the CTR of an ad. To make the empirical study more convinced, we compare different algorithms based on the real revenue obtained in the test data. The revenue is calculated by summing up the bidding price from the clicked ads. Figures 3(b), (d), (f) and (h) shows the performances of the three algorithms. It could be observed that the real case is indeed different from the estimated case. In the estimated case, the error is only around 5 %; but in the real case, when λ is small, the error can be very large. This is also expected, because the CTR of ads in general is very small, thus the variance is high when λ is small. From the figure, the comparisons among the three algorithms follow the same pattern with the previous one. The dynamic algorithm consistently outperforms the static one; and the combination algorithm consistently outperforms the dynamic one. The average improvement of the combination method over the static method is 12.3 % with the exact-match dataset, and 6.76 % with the broad-match dataset. In sponsored search, such a distance means billions of US dollars per year for search engine agencies. It is a significant improvement.

6 Related Work

Previous related work is mainly from the field of information retrieval. The most common method is to utilize the CTR prediction of a query-ad pair. When the CTR is predicted, the revenue can be calculated by multiplying it by its bidding price. The methods of estimating CTR can be divided into click models [4], and regression models [3]. In addition, some classification-based methods are also proposed to decide whether or not the ads should be displayed [1,9,10,12].

7 Conclusion

In this paper, as a primary study to measure the commercial load in sponsored search, we propose to quantify it by the average displayed ad number per query. And then we investigate how to optimize the displaying strategy in the constraint of a certain commercial load. The optimization problem is formalized in the framework of the secretary problem. By extending previous algorithm for the multi-choice secretary problem, we propose a novel dynamic algorithm for this task. Through theoretical analysis, we demonstrate that the expectation performance is approaching the optimal solution; and by experimental verifications, our proposed algorithm has a significant improvement over the baselines.

Acknowledgements. The work described in this paper was fully supported by National Natural Science Foundation of China (No. 61300076) and Ph.D. Programs Foundation of Ministry of Education of China (No. 20131101120035).

References

1. Broder, A., Ciaramita, M., Fontoura, M., Gabrilovich, E., Josifovski, V., Metzler, D., Murdock, V., Plachouras, V.: To swing or not to swing: learning when (not) to advertise. In: Proceedings of CIKM'08, pp. 1003–1012. ACM (2008)
2. Buscher, G., Dumais, S.T., Cutrell, E.: The good, the bad, and the random: an eye-tracking study of ad quality in web search. In: Proceedings of SIGIR'10, pp. 42–49. ACM (2010)
3. Graepel, T., Candela, J.Q., Borchert, T., Herbrich, R.: Web-scale bayesian click-through rate prediction for sponsored search advertising in microsofts bing search engine. In: Proceedings of ICML'10 (2010)
4. Guo, F., Liu, C., Kannan, A., Minka, T., Taylor, M., Wang, Y.M., Faloutsos, C.: Click chain model in web search. In: Proceedings of WWW'09, pp. 11–20. ACM (2009)
5. Jansen, B.J., Resnick, M.: An examination of searcher's perceptions of nonsponsored and sponsored links during ecommerce web searching. J. Am. Soc. Inf. Sci. Technol. **57**(14), 1949–1949 (2006)
6. Kleinberg, R.: A multiple-choice secretary algorithm with applications to online auctions. In: Proceedings of ACM-SIAM Symposium on Discrete algorithms, pp. 630–631 (2005)
7. Lee, W.: Preference strength, expected value difference and expected regret ratio. Psychol. Bull. **75**(3), 186 (1971)
8. Marable, L.: False oracles: consumer reaction to learning the truth about how search engines work, results of an ethnographic study (2003)
9. Nath, A., Mukherjee, S., Jain, P., Goyal, N., Laxman, S.: Ad impression forecasting for sponsored search. In: Proceedings of WWW'13, pp. 943–952. ACM (2013)
10. Pandey, S., Punera, K., Fontoura, M., Josifovski, V.: Estimating advertisability of tail queries for sponsored search. In: Proceedings of SIGIR'10, pp. 563–570 (2010)
11. Radovanovic, A., Heavlin, W.D.: Risk-aware revenue maximization in display advertising. In: Proceedings of WWW'12, pp. 91–100. ACM (2012)
12. Zhu, Y., Wang, G., Yang, J., Wang, D., Yan, J., Hu, J., Chen, Z.: Optimizing search engine revenue in sponsored search. In Proceedings of SIGIR'09, pp. 588–595. ACM (2009)

Models for Distributed, Large Scale Data Cleaning

Vincent J. Maccio, Fei Chiang(✉), and Douglas G. Down

McMaster University, Hamilton, ON, Canada
{macciov,fchiang,downd}@mcmaster.ca

Abstract. Poor data quality is a serious and costly problem affecting organizations across all industries. Real data is often dirty, containing missing, erroneous, incomplete, and duplicate values. Declarative data cleaning techniques have been proposed to resolve some of these underlying errors by identifying the inconsistencies and proposing updates to the data. However, much of this work has focused on cleaning data in static environments. Given the Big Data era, modern applications are operating in dynamic data environments where large scale data may be frequently changing. For example, consider data in sensor environments where there is a frequent stream of data arrivals, or financial data of stock prices and trading volumes. Data cleaning in such dynamic environments requires understanding the properties of the incoming data streams, and configuration of system parameters to maximize performance and improved data quality. In this paper, we present a set of queueing models, and analyze the impact of various system parameters on the output quality of a data cleaning system and its performance. We assume random routing in our models, and consider a variety of system configurations that reflect potential data cleaning scenarios. We present experimental results showing that our models are able to closely predict expected system behaviour.

Keywords: Data quality · Distributed data cleaning · Queueing models

1 Introduction

In the Big Data era, data quality has become a prolific issue spanning across all industries. Data-driven decision making requires having access to high quality data that is clean, consistent and accurate. Unfortunately, most real data is dirty. Inconsistencies arise due to the integration of data from multiple, heterogeneous data sources, where each source has its own data representation and format. Data inconsistencies also arise when integrity constraints, the rules defined over the data to keep the data accurate and consistent, are violated and not strictly enforced. When integrity constraints are not strongly enforced, there is no mechanism to validate whether the data updates are correct, thereby leading to data errors.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 369–380, 2014.
DOI: 10.1007/978-3-319-13186-3_34

Existing data cleaning systems [1–7] have proposed techniques to repair the data by suggesting modifications to the data that conform to user expectations or to a given set of integrity constraints. However, most of these systems have focused on cleaning *static* instances of the data. That is, a snapshot of the data is taken, and if any changes in the data occur after the snapshot, then a new snapshot must be taken. While existing techniques may work well for static data environments, many modern applications are now operating in dynamic data environments where the data is frequently changing. For example, data in sensor, financial, retail transactions, and traffic environments require processing large scale data in near real-time settings. This has led to the need for more dynamic data cleaning solutions [8], particularly, those that can support large scale datasets.

In this paper, we present a set of models for distributed large scale data cleaning, where the data cleaning task is delegated over a set of servers. Such models are relevant for cleaning large data sets where the data can be partitioned and distributed in a parallel server environment. Each server has its own properties, such as the service time in which it cleanses the data, and the quality of the output data. We apply principles from queueing theory to model a variety of server configurations, and analyze the tradeoff between the system performance and data quality. We consider variations in our model, such as job priorities, and servers discarding data. Finally, we present our experimental results showing the accuracy of our model predictions against simulation results using real datasets.

This paper is organized as follows. In Sect. 2, we present related work, followed by preliminary background in Sect. 3. In Sect. 4, we present details of our distributed data cleaning model, and the variations we consider that reflect real application environments. Finally, we present our experimental results in Sect. 5, and conclude in Sect. 6.

2 Related Work

Recent data cleaning systems such as AJAX [2], Nadeef [3], LLUNATIC [9] and others, have focused on cleaning a static snapshot of the data for a given set of constraints. While these solutions are effective for data environments where the data and the constraints may not change frequently, they are expensive to implement in environments where the data and the constraints may evolve, as they require manual re-tuning of parameters, and acquisition of new data and constraints. In our work, we focus on modelling distributed data cleaning in dynamic environments, and study the influence of parameter changes on the data quality output.

As data intensive applications increasingly operate in data environments where rules and business policies may change, recent work in this area has proposed repair techniques to consider evolving constraints, primarily focused on functional dependencies (FDs) [10,11]. However, the objective of these techniques is to propose specific modifications to the data and/or constraints to resolve the inconsistency. Given the increasing need for scalable data cleaning systems, our objective is to analyze the behaviour of such a system under different parameter settings and configurations. To the best of our knowledge, none of the existing work has considered this.

3 Preliminaries

In this section, we provide a brief background on simple queueing models. A popular queueing model is the M/M/1 queue. This is used to represent a single server system with the following characteristics. First, jobs arrive to the system according to a Poisson process with rate λ, that is, the time between arriving jobs is exponentially distributed with rate λ. Second, jobs that arrive to the system are served one at a time following a First In First Out (FIFO) policy. Lastly, the time that it takes for a job to be processed by the server, often referred to as the *service time*, is exponentially distributed with rate μ. We refer to response time as the time from arrival to departure of a job in the system, and *service time* as the time to service (clean) a job once it enters the server. The two M's in the M/M/1 notation denote the characteristics of the arrival process and service time distribution, respectively, and represent for Markovian (or memoryless), while the 1 denotes that it is a single server system.

We can analyze this system as a Continuous Time Markov Chain, and expressions for the expected number of jobs in the system, $\mathbb{E}[N]$, as well as the expected response time, $\mathbb{E}[R]$, can be given by [12]:

$$\mathbb{E}[N] = \frac{\lambda}{\mu - \lambda} \quad \text{and} \quad \mathbb{E}[R] = \frac{1}{\mu - \lambda}. \tag{1}$$

The assumption of the exponential service times in the M/M/1 model can be limiting in some situations. In the M/G/1 queue, the service times follow a general distribution, G, where the first and second moments are known. Similar to the M/M/1 queue, closed form expressions for $\mathbb{E}[N]$ and $\mathbb{E}[R]$ are given by:

$$\mathbb{E}[N] = \rho + \frac{\rho^2 + \lambda^2\sigma_S^2}{2(1-\rho)} \quad \text{and} \quad \mathbb{E}[R] = \frac{1}{\mu} + \frac{\rho^2 + \lambda^2\sigma_S^2}{2\lambda(1-\rho)}, \tag{2}$$

where σ_S^2 denotes the variance of the service time distribution. More details about these and related models can be found in [12–14].

4 Distributed Data Cleaning

We present models for large scale data cleaning in dynamic data environments. We assume that there is some baseline data instance, and changes to the data are reflected by incremental updates to the baseline instance. These incremental updates occur with some frequency, which we call the *arrival rate*. As this data arrives, there is a data cleaning task, a *job*, that needs to be done using this data (in conjunction with the baseline data instance). Given a set of servers (each with its own properties), and a set of data arrival streams, what is the optimal assignment of incoming data jobs to servers, such that the data quality output is maximized and the overall system performance is stable?

To solve this problem, we assume several random routing settings, where each server is independent. Furthermore, the decision of which server handles an incoming data job is made independently of the current state of the servers. If we do not make such assumptions, determining the optimal configuration would be intractable as it would be an extension of the "slow server problem" [15], a well-known problem in the queueing literature. We begin by considering a simple base model with two servers, and extend this model to consider other variations. For each variation, we provide analytic results expressing the performance vs. data quality tradeoff, and provide our insights on system behaviour. We also present our experimental results comparing our predicted model values against a simulation using a real energy metering data stream. We believe our work provides meaningful insights to a data analyst on the expected system behaviour and quality of the data cleaning task for large scale data.

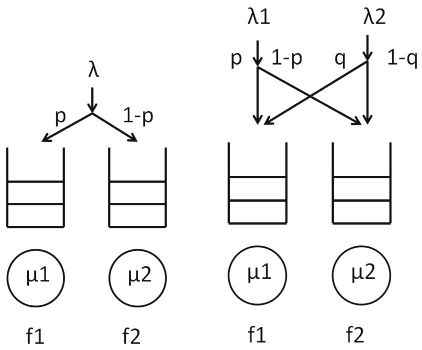
4.1 The Base Model

Consider a data cleaning system with two distinct servers, where jobs arrive according to a Poisson process with rate λ. When a job arrives, it is sent to the first server with probability p, and to the second server with probability $(1-p)$. From the demultiplexing of the arrival stream via routing, each server can be seen as an independent M/M/1 queue. Furthermore, each server has an associated failure probability, denoted f_1 and f_2, where $0 \leq f_1, f_2 \leq 1$. These values represent the probability that a server incorrectly cleaned the data. That is, after a job completes, the server either failed or the proposed updates to the data were incorrect. We define a random variable F, such that $0 \leq F \leq 1$, that denotes the proportion of jobs in the system that have been incorrectly cleaned in steady state, and the data remains dirty. Let $\mathbb{E}[F]$ denote the expected value of F. Figure 1(a) shows the base model.

We assume $\lambda < \mu_1 + \mu_2$ to ensure system stability, and that the user cannot control λ, μ_1, μ_2, f_1 nor f_2, but can set the value for the routing probability p. Since the user has control of p, the goal is to select a value for p that minimizes expected response time, and maximizes the expected data quality (by minimizing $\mathbb{E}[F]$). We define a cost function to analyze this tradeoff.

$$\mathcal{C}_{Base} = \mathbb{E}[R] + \beta\mathbb{E}[F] \tag{3}$$

where the value β represents the relative weight of the expected data quality over the expected response time. We assume β is given by the user. The cost $\mathcal{C}_{Base}$ models our objective to minimize system response time, and to minimize failure (poor data quality results). We consider $\mathcal{C}_{Base}$ as an initial cost function. As future work, we plan to extend $\mathcal{C}_{Base}$ to include more complex, constrained models that minimize $\mathbb{E}[R]$ subject to $\mathbb{E}[F]$ satisfying minimum threshold levels. We can determine closed form expressions for $\mathbb{E}[R]$ and $\mathbb{E}[F]$ by noting that the



(a)Twoserver routing

(b)Classesandpriorities

Fig. 1. Graphical queueing models

inputs to the two queues are a demultiplexed Poisson process, and therefore can be viewed as M/M/1 queues. By applying Eq. (1) we obtain:

$$\mathbb{E}[R] = \frac{p}{\mu - p\lambda} + \frac{1-p}{\mu - (1-p)\lambda} \tag{4}$$

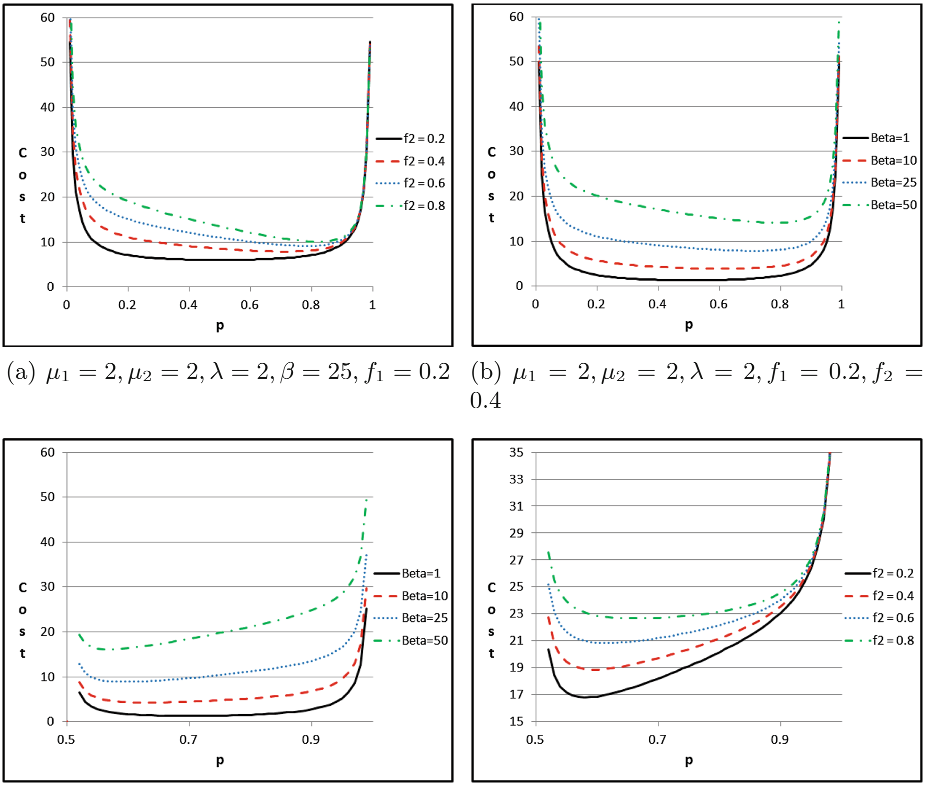
$$\mathbb{E}[F] = pf_1 + (1-p)f_2. \tag{5}$$

Equations (4) and (5) can be applied to compute expected response time and expected data quality output, respectively. Without loss of generality, such closed form expressions can also be computed for more complex, constrained models as mentioned above. Substituting (4) and (5) into (3) yields a closed form cost function.

$$\mathcal{C}_{Base} = \beta(pf_1 + (1-p)f_2) + \frac{p}{\mu - p\lambda} + \frac{1-p}{\mu - (1-p)\lambda}. \tag{6}$$

Unfortunately, taking the derivative of (6), and setting it to zero does not yield a closed form expression in terms of p. However, we can analyze the tradeoff between p and the remaining parameters in (6) by plotting the cost $\mathcal{C}_{Base}$ against p for several configurations, as shown in Fig. 2. We will refer to the optimal value of p as p^*.

Figure 2(a) and (b) show two system configurations where each system has data jobs arriving at the same rate in which they can be processed ($\lambda = \mu_1 = \mu_2$). In Fig. 2(a), the second server produces lower quality data than the first server ($f_1 \leq f_2$). In Fig. 2(b), there is a greater weight (β) on improved data quality than system response time. In both Fig. 2(a) and (b), p should be selected at the point where the cost is minimal. We observe that in both graphs, the cost is fairly low for a wide range of p values, ranging from [0.1, 0.9], indicating that both systems are fairly stable. It is only at the endpoints ($p \to 0, p \to 1$) where the cost sharply increases and the system becomes unstable. Hence, a conservative



(a) $\mu_1 = 2, \mu_2 = 2, \lambda = 2, \beta = 25, f_1 = 0.2$ (b) $\mu_1 = 2, \mu_2 = 2, \lambda = 2, f_1 = 0.2, f_2 = 0.4$

(c) $\mu_1 = 4, \mu_2 = 2, \lambda = 4, f_1 = 0.5, f_2 = 0$ (d) $\mu_1 = 4, \mu_2 = 2, \lambda = 4, \beta = 25, f_1 = 0.9$

Fig. 2. Base model configurations

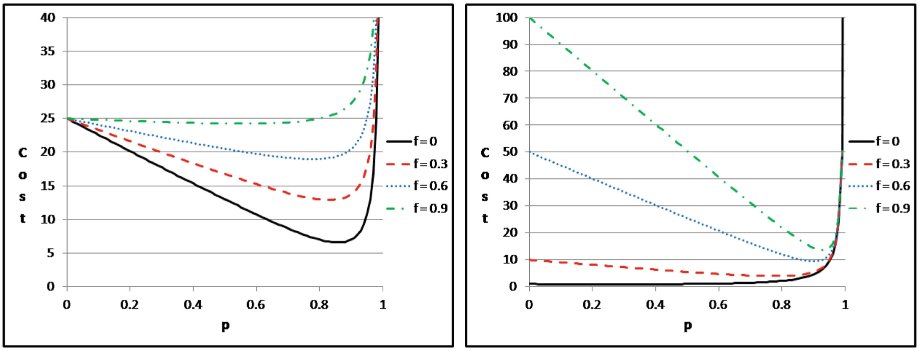
approach is to assign arriving data jobs equally to both servers ($p = 0.5$) to maintain system stability.

Figure 2(c) and (d) show less stable systems (than Fig. 2(a) and (b)), where selecting a different p value yields more dramatic effects on the cost, and ultimately in the system response times. Selecting p must be done carefully to minimize the cost, as can be seen in Fig. 2(d), where for $p > 0.6$, steeper curves reflect increasing instability in the system.

4.2 The Discard Model

In this section, we consider the case where a server may choose to *discard* particular data jobs. We consider discarding jobs for two reasons:

1. If the overall system is overloaded ($\lambda > \mu_1$), then arriving data jobs must be discarded to resume stability.
2. When service level agreements (SLAs) must be met, some incoming data jobs may need to be discarded.



(a) $\mu = 2, \lambda = 2, \beta = 25$

(b) $\mu = 2, \lambda = 2, f = 0$

Fig. 3. Discard model configurations

At the routing step, if the decision is made to discard a job, then the failure rate $f = 1$, and the response time of the discarded job is instantaneous. There is no longer a restriction on λ to ensure stability, as incoming jobs can simply be removed from the system until the server is able to handle the incoming data. The discard model can be viewed as an instantiation of the base model (described in Sect. 4.1), where $f_2 = 1$ and $\mu_2 \to \infty$. The resulting cost function is,

$$\mathcal{C}_{Discard} = \beta(p(f_1 - 1) + 1) + \frac{p}{\mu - p\lambda} \tag{7}$$

Figure 3(a) shows a system at full capacity ($\mu = \lambda$). The value f is varied from 0 (where the system always produces clean data and jobs are not discarded) to 0.9 (where most of the data will be incorrectly cleaned). We observe that for $f = 0$, with only one server, the system is quite unstable as seen by the wide fluctuation in cost values for $p \in [0.1, 0.9]$. As $f \to 0.9$, the system performance improves as incoming data jobs may be discarded to achieve stability.

In Fig. 3(b), we investigate the influence of β on the overall system performance (recall β is a weight of the relative importance between data quality vs. system response time). As β ranges from [1,100], we want the system to produce increasingly higher quality output. For increasing β values, the system becomes increasingly unstable, as shown by the increasingly steep (negative sloped) curves. This indicates that in such a single server system, if we want to achieve improved data quality, we must be willing to tolerate wide fluctuations in system response time and stability.

4.3 Classes and Priorities

The models we have considered thus far include a single data arrival stream, and follow a FIFO policy. In this section, we explore how adding another data arrival stream, and including priorities influence the system behaviour. We consider two incoming data streams, where jobs still follow a FIFO policy. The data jobs arrive to each server at a rate of λ_1 and λ_2, and both are assumed to be Poisson processes. Each of the servers service incoming jobs following an exponential distribution, but depending on which stream the job arrived from, it may process the jobs with different service rates. We consider this preferential notion to model *priority* between data jobs, which exists in many applications systems to guarantee service level agreements.

For simplicity, we assume that the two servers are homogeneous with respect to their processing rates. Therefore, μ_1 and μ_2, denote the service rate of data jobs arriving from the first and second stream, respectively. We let p and q represent the probability that a data job is sent to the first server from the first and second stream, respectively. Similarly, an incoming job is sent to the second server with probability $(1-p)$ and $(1-q)$, from the first and second streams, respectively. Hence, a data analyst must determine appropriate values for p and q to maximize the data quality output from the servers, and minimize system response time. The system can be seen in Fig. 1(b).

Our analysis here differs from prior models as we no longer have two M/M/1 queues, due to the differing service rates μ_1 and μ_2. However, we observe that while the service time distribution is not exponential, it is hyper-exponential. Therefore, an M/G/1 queue can be fitted to the system, and the Pollaczek-Khinchine formula, (2), can be used. This leads to the following cost function,

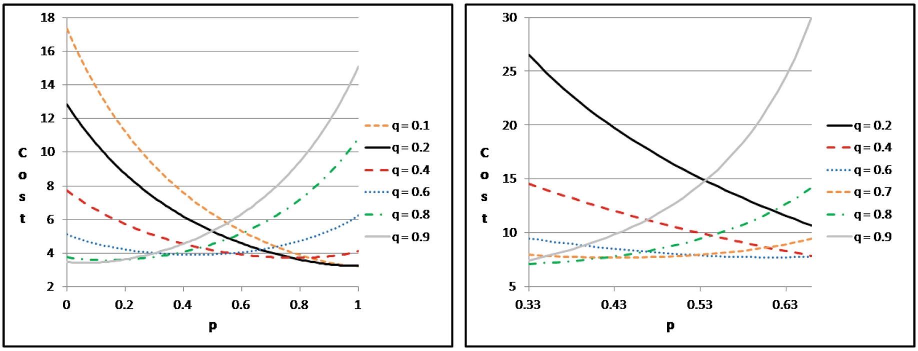
$$\begin{aligned}\mathcal{C}_{Classes} = \beta((p+q)f_1 + (2-p-q)f_2) + (p+q)\left(\frac{\rho_1}{\lambda_1^*} + \frac{\rho_1^2 + (\lambda_1^*)^2\sigma_{S1}^2}{2\lambda_1^*(1-\rho_1)}\right)\\ + (2-p-q)\left(\frac{\rho_2}{\lambda_2^*} + \frac{\rho_2^2 + (\lambda_2^*)^2\sigma_{S2}^2}{2\lambda_2^*(1-\rho_2)}\right),\end{aligned}$$

where,

$$\begin{aligned}\lambda_1^* &= p\lambda_1 + q\lambda_2\\ \lambda_2^* &= (1-p)\lambda_1 + (1-q)\lambda_2\\ \rho_1 &= \frac{\lambda_1^*}{\frac{p\lambda_1}{p\lambda_1+q\lambda_2}\mu_1 + \frac{q\lambda_2}{p\lambda_1+q\lambda_2}\mu_2}\\ \rho_2 &= \frac{\lambda_1^*}{\frac{(1-p)\lambda_1}{(1-p)\lambda_1+(1-q)\lambda_2}\mu_1 + \frac{(1-q)\lambda_2}{(1-p)\lambda_1+(1-q)\lambda_2}\mu_2}\\ \sigma_{S1}^2 &= \left(\frac{p}{\mu_1} + \frac{q}{\mu_2}\right)^2 + 2pq\left(\frac{1}{\mu_1} - \frac{1}{\mu_2}\right)^2\\ \sigma_{S2}^2 &= \left(\frac{1-p}{\mu_1} + \frac{1-q}{\mu_2}\right)^2 + 2(1-p)(1-q)\left(\frac{1}{\mu_1} - \frac{1}{\mu_2}\right)^2.\end{aligned}$$

The physical interpretation of these parameters are: λ_1^* and λ_2^* are the arrival rates, ρ_1 and ρ_2 are the utilizations, and σ_{S1}^2 and σ_{S2}^2 are the variances of the service time distributions, for the first and second server, respectively. These follow from Eq. (2), and the definitions of the hyper-exponential distribution [12].

Using the above formulas, we plot a set of configurations as shown in Fig. 4. In both Fig. 4(a) and (b), the optimal parameter setting has a small value for q



(a) $\mu_1 = 6, \mu_2 = 1.5, \lambda_1 = 2, \lambda_2 = 1 f_1 = 0.05, f_2 = 0.2, \beta = 25$

(b) $\mu_1 = 4, \mu_2 = 1.25, \lambda_1 = 2, \lambda_2 = 1 f_1 = 0, f_2 = 0.5, \beta = 5$

Fig. 4. Modelling with priorities

(approximately 0.2), but a large value for p (approximately 0.9). In both configurations, the first server is more reliable and produces higher quality data ($f_1 < f_2$). Our recommendation in this case is to assign incoming jobs to the first server without overloading it, so as to minimize the cost and maintain system stability. The remaining jobs can be sent to the second server so that it does not remain idle.

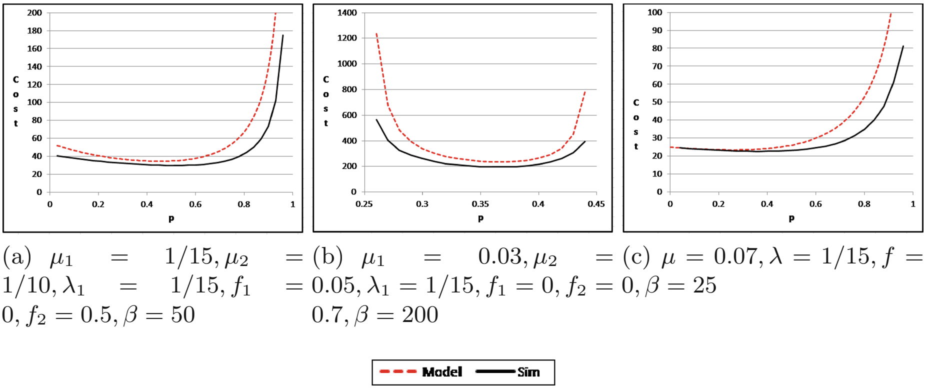
We now consider priorities between the incoming data streams, also referred to as *classes*. Although the analysis becomes more difficult, it remains tractable. Consider the system in Fig. 1(b), where jobs that arrive from the first stream have higher priority than (i.e., they preempt) jobs from the second stream. Furthermore, if a high priority job arrives to the system while a low priority job is being processed, it preempts its execution, and takes its place.

We assume two priority classes: low and high. Each arriving high priority job views the system as a simple M/M/1 queue where only high priority jobs exist, with arrival rate $p\lambda_1$. We restrict the two job classes to share the same service time distribution, relative to the server. That is, μ_1 represents the service rate, for jobs of both classes, at the first server, while μ_2 represents the service rate for jobs of both classes at the second server. Since traditional M/M/1 queues can now be applied, we view the arrival rate as $p\lambda_1 + q\lambda_2$. Interestingly, we can derive expressions for the number of jobs in the system, for each priority class, as given below.

$$\mathbb{E}[N] = \mathbb{E}[N_{high}] + \mathbb{E}[N_{low}]$$

where $\mathbb{E}[N]$ denotes the number of jobs in a specific queue. Due to the previous observation that high priority jobs can be modelled as an M/M/1 queue with arrival rate $p\lambda_1$, we can derive the number of low priority jobs at the first queue.

$$\mathbb{E}[N_{low}] = \frac{p\lambda_1 + q\lambda_2}{\mu_1 - p\lambda_1 - q\lambda_2} - \frac{p\lambda_1}{\mu_1 - p\lambda_1}. \tag{8}$$



(a) $\mu_1 = 1/15, \mu_2 = 1/10, \lambda_1 = 1/15, f_1 = 0, f_2 = 0.5, \beta = 50$

(b) $\mu_1 = 0.03, \mu_2 = 0.05, \lambda_1 = 1/15, f_1 = 0, f_2 = 0.7, \beta = 200$

(c) $\mu = 0.07, \lambda = 1/15, f = 0, \beta = 25$

Fig. 5. Experiments

Equation 8 allows us to compute the expected number of low priority jobs in the system. In addition to the expression for $\mathbb{E}[N]$, we can specify constraints on the system that limit the total number of (low priority) jobs. For example, if we have SLAs, we can impose constraints on the system that allow a proportion of the servers to only handle low priority jobs. Furthermore, given the arrival rate, service rate, and probabilities p and q, we can directly compute the estimated number of jobs per class, which can enable a data analyst to predict system behaviour. In addition, we can also consider how the server failure rates, per job class priority, will be affected by changes in these parameters. For example, high priority jobs should be serviced by reliable servers where f is close to 0. We plan to investigate this direction as future work.

5 Experiments

In this section, we validate the accuracy of our proposed models by comparing the optimal routing probabilities (p^*), to the values given by our simulated system. In our simulations, we assume our data arrival streams follow a Poisson process. We simulated 100,000 arriving jobs using the CSIM 19 library [16]. Our simulation was run on a server with Intel Xeon E5-2960 processors, 4 vCPUs, and 16 GB of RAM. We used real data describing energy usage across our University campus, that reported values such as water, hydro usage, and air temperature readings. The readings are reported every 15 min.

Figure 5 shows our simulation results. Figure 5(a) shows a lightly loaded system where the base model is used. We observe that the shape of the curves are quite similar, and as expected, the cost from the simulation is less than that of the model. The value of p^*, predicted by our model is within 10 % of the simulated p value. If we chose to use the predicted routing probability from the queueing model, our results in the simulated system would yield a cost that is within 10 % of the minimum. We note that the difference in p^* values occurs in the insensitive portion of the curve ($p \in [0.05, 0.4]$) where the cost is relatively constant, and the model predictions closely follow the simulation results. Our results in Fig. 5(b) show very promising results. Figure 5(b) shows a heavily loaded base model system, which can become unstable depending on the choice of p. We found that the values of p^* predicted by the queueing and simulation models were equal.

In our last experiment, we simulated the discard model. Figure 5(c) shows that the values of p^* from the queueing and simulations models differ by approximately 0.08. If we apply the values predicted by the queueing model in our system, our cost would be within 2.5 % of the minimal cost for this workload. Overall, our evaluation has provided very promising initial results showing that our models are able to closely predict optimal routing probabilities for incoming data cleaning tasks. By having these optimal values, data analysts are able to better understand and predict anticipated system load, and stability, in order to maximize the data quality output from the system.

6 Conclusion

In this paper, we have taken a first step towards modelling a distributed data cleaning environment where each job represents the incoming data that needs to be cleaned with respect to a baseline data instance. We have presented a set of models that reflect different data cleaning system configurations. Specifically, we have presented analytic models, expressions, and insights for a base model, discard model (where jobs can be discarded), and a priority-class model. We have investigated the stability (and response time) to data quality tradeoff for each of these models, and revealed some interesting insights. For example, we have observed that particular configurations can lead to unstable system performance, and provided cases when discarding incoming jobs may be necessary. Our evaluation revealed promising accuracy results, where our model predictions are all within 10 % of the simulated results. Avenues for future work include further model validation, extending the models to consider n servers, and investigating constrained models where the parameters are subject to a set of cost or threshold constraints.

References

1. Raman, V., Hellerstein, J.M.: Potter's wheel: An interactive data cleaning system. In: VLBD, pp. 381–390 (2001)
2. Galhardas, H., Florescu, D., Shasha, D., Simon, E., Saita, C.A.: Declarative data cleaning: Language, model, and algorithms. In: VLDB, pp. 371–380 (2001)
3. Dallachiesa, M., Ebaid, A., Eldawy, A., Elmagarmid, A., Ilyas, I.F., Ouzzani, M., Tang, N.: NADEEF: a commodity data cleaning system. In: SIGMOD, pp. 541–552 (2013)
4. Bohannon, P., Fan, W., Flaster, M., Rastogi, R.: A cost-based model and effective heuristic for repairing constraints by value modification. In: SIGMOD, pp. 143–154 (2005)

5. Yakout, M., Elmagarmid, A.K., Neville, J., Ouzzani, M., Ilyas, I.F.: Guided data repair. VLDB Endow. **4**(5), 279–289 (2011)
6. Chiang, F., Miller, R.J.: A unified model for data and constraint repair. In: ICDE, pp. 446–457 (2011)
7. Chiang, F., Wang, Y.: Repairing integrity rules for improved data quality. IJIQ 20 p. (2014)
8. Volkovs, M., Chiang, F., Szlichta, J., Miller, R.J.: Continuous data cleaning. In: ICDE, pp. 244–255 (2014)
9. Geerts, F., Mecca, G., Papotti, P., Santoro, D.: The LLUNATIC data-cleaning framework. PVLDB **6**(9), 625–636 (2013)
10. Chiang, F., Miller, R.J.: Active repair of data quality rules. In: ICIQ, pp. 174–188 (2011)
11. Beskales, G., Ilyas, I.F., Golab, L., Galiullin, A.: On the relative trust between inconsistent data and inaccurate constraints. In: ICDE, pp. 541–552 (2013)
12. Gross, D., Harris, C.M.: Fundamentals of Queueing Theory, 3rd edn. Wiley-Interscience, New York (1998)
13. Harchol-Balter, M.: Performance Modeling and Design of Computer Systems: Queueing Theory in Action. Cambridge University Press, New York (2013)
14. Kleinrock, L.: Queueing Systems, vol. 1. Wiley-Interscience, New York (1975)
15. Rubinovitch, M.: The slow server problem. J. Appl. Probab. **22**(4), 205–213 (1985)
16. Mesquite Software CSIM 19. http://www.mesquite.com/

A Study of Rumor Spreading with Epidemic Model Based on Network Topology

Dawei Meng, Lizhi Wan, and Lei Zhang(✉)

Graduate School at Shenzhen, Tsinghua University,
Shenzhen, People's Republic of China
{mengdaweil16, onelizhi}@gmail.com,
zhanglei@sz.tsinghua.edu.cn

Abstract. The history of rumors might be as long as the history of human beings. Not much is known about this phenomenon until very recently when social networks provide a more efficient way for online rumor spreading. Microblogging platform, like Twitter and Sina Weibo, provides an ideal environment not only for rumor spreading, but also fruitful data to reveal the underlying rules controlling the birth, spread, and death of rumors. In this paper, we try to answer the following questions which have been confusing people for years. How do rumor propagate in the network? How do human factors affect the spreading patterns of rumor? How do we construct models to understand the collective group behavior based on probabilistic individual choices? Our study on microblogging systems help to expose the digital traces of online rumor spreading, which offers an opportunity to investigate how netizens interact with each other in the process of rumor spreading. Based on the analysis of the interactions, we propose a mathematical model running on real topologies to simulate this process and explore the characteristics of online rumor spreading. Experiments show that our model can reflect the transmission pattern of real rumors. Our work is different from previous ones by pointing out that a subset of the population can work as "Resisters" who are proactively persuading their neighbors not to believe the rumor, while previous works only label them as the "silent" nodes. To the best of our knowledge, this is the first time to emphasize the contribution of "Resisters" and to mathematically study their influence on the rumor spreading process.

Keywords: Social network analysis · Information spread pattern · SISR model

1 Introduction

Rumors spread more quickly and widely on the web than traditional ways. According to a statistical report released by China Internet Network Information Center (CNNIC), online rumors have become the main source of rumors and most of them come from the Weibo platform [7]. As a result, it is more and more difficult to identify the source of rumors. Moreover the mutation of rumors is beyond the capture of authority. When the authority starts to refute the rumors, the misinformation has been widely known to the public. It seems impossible to clarify rumors promptly.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 381–392, 2014.
DOI: 10.1007/978-3-319-13186-3_35

Despite the rumors are in essence unconfirmed messages, if they are spread in a large scale, the harm is still huge. One example is the rumor about the death of Liuxiao Lingtong, who is the actor of Monkey King. On March 12, 2013, some netizens released the news that Liuxiao Lingtong died of illness on the microblogging. The news sparked the remembrance of a generation of people on the classic image of Monkey King. This microblogging was forwarded in the absence of a proven. The next day at noon, Liuxiao Lingtong posted a statement on microblogging, saying that he encounter "been dead", which proved that the previous microblogging was false message. However, there were still people who didn't know the truth believed that "Liuxiao Lingtong" was dead, because they had seen the microblogging forwarded by others. This false message not only brought reputational damage to Liuxiao Lingtong, but also brought shock and psychological examination of the truth to his fans.

Besides the influence to individuals, rumors could also affect the socio-economic activities. A typical example is the "salt panic" after the Japan nuclear crisis. In March 2011, the earthquake in Japan caused the Fukushima nuclear leak. After the nuclear leakage incident, someone claimed that salt can prevent nuclear radiation, the news spread on the network, and a wave of "salt panic" tide was set off in China. Within only a few days, salt in most major supermarkets was sold out. More unscrupulous businessmen took the chance to raise the price. The price of a pack of salt which was a few dollars at first was fired up to dozens. This "salt panic" storm even affected the stock market, and salt-related stocks showed a rising trend as well. Then there were comments that this "panic buying salt" storm was operated by someone on the behind. With the "panic buying salt" storm getting worse, the news that "salt can prevent radiation" was spread more and more widely, the government came out to refute the rumor via authoritative media, and at the same time increased market supply of salt. Finally, the situation gradually stabilized, and the rumor was put out. Rumors' harm could be seen by these examples.

Therefore, it is necessary to build a mathematical model for rumor spreading for the following purposes: First, to acquire a better understanding of the structure and evolution procedures of rumor spread; Second, to provide a scheme to evaluate the power of rumor spread and to estimate the power grown trend in the future; Third, to identify the parameters affecting the process of a rumor spread. How to model rumor spread in the environment of web2.0 is the main concern of this paper.

The rest of the paper is organized as follows: Sect. 2 gives a brief survey on the research work been done on rumor spread. Section 3 would introduce our mathematical model in detail. The real life rumor data is explained and the mathematical model is used to simulate the data in Sect. 4. Finally Sect. 5 concludes the paper and points out possible future work directions.

2 Related Work

Currently, many rumor models have been proposed. The Daley–Kendall (DK) model is the earliest model which could be identified. DK model proposed a rumor framework by which almost all later rumor models followed [1]. Firstly, this model divided the target population into three groups, each group with a unique state. The three states are *ignorant*, *spreader* and *stifler*. *Ignorant* are those who have never heard of the rumor before. *Spreaders* are those who have heard the rumor and are willing to spread it to their neighbors. Thus making its *ignorant* neighbors turn into *spreaders*. *Stiflers* are those who also hear rumors but refuse to spread it. *Stiflers* act as the black hole in the process of rumor transmission. DK model suggested that a *spreader* would infect an *ignorant* node through pairwise contact, thus making an *ignorant* node change into a *spreader*. When a *spreader* encounters another *spreader*, both *spreader* would turn into *stifler* because of the staleness of the rumor information. When a *stifler* meets a *spreader*, the latter would turn into a *stifler*. DK model assumes that the whole population is fixed and no death or birth of nodes is considered.

In a later time, a more widely used model named Maki–Thompson (MT) rumor model is introduced. It was proposed by Maki-Thompson. It made a change into the contact rule that when two spreaders meet, only the spreader who initiates the contact would change its state [2]. After that a stochastic MT model was proposed to solve the stochastic process, which means that when a spreader encountered an ignorant, the latter would not become a spreader in a deterministic way but with a probability instead. By applying probability theory into the MT model makes it fit into reality in a better way. After that, many models based on MT model were introduced including how to further divide spreaders into active spreaders and inactive spreaders etc.

Nekovee [3] and Zanette [4] researched rumor spread in small-world network based on MT model and found that the propagation of rumor was closely related to the underlying network. Moreno [5] studied the rumor propagation characteristics and conclude that the uniformity of network had significant impact on the spread of rumors. R. Thompson et al. [6] proposed a modified model on the basis of DK model which subdivided S and I. X. Wang [8] proposed that the increase of clustering coefficient could effectively inhibit the spread of rumors. Singh et al. [9] studied how to monitor and suppress the spread of rumors by placed observer in the small-world networks. Zanette et al. [10] proposed a rumors immunization strategy in small-world networks and pointed that rumors could be limited in one group by employing certain immune measures. Budak et al. [11] studied how to reduce the influence of rumors by using a resist process to spread negative information. Tripathy et al. [12] proposed two strategies to control the spread of rumors: one was to delay resistance, another was to place some monitoring nodes on fixed location in the network.

3 The SISR Model

3.1 Dynamics of SISR Model

Although MT model is widely used in research of rumor spread, the notation of MT model is somewhat confusing. As the epidemic model SIR model is similar to MT model, we use the annotation of the SIR model to explain the proposed model. In SIR model [13], *S* stands for susceptible, which means those who are vulnerable for rumors. *I* stands for infected, which means those who believe in the rumor and thus become infectious. *R* stands for recovered, which means those who do not believe the rumor and would not transmit it.

The notations in SIR model describe the rumor spread in a more clear way. We use the notation for SISR model. Here *S* represents susceptible and *I* represents infected, but *R* here represents resistant. In our model, the resistant not only distrust the rumor, but also refute it. Figure 1 shows the transition pattern of SISR model.

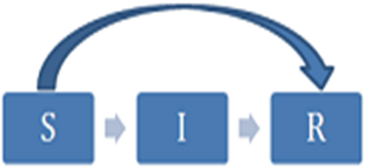


Fig. 1. Transition pattern of SISR model

The most distinguished change of SISR model is that a shortcut from susceptible to resistant is added, meaning that those who have not heard the rumor can choose not to believe when they first heard it. The dynamics of infected turning into resistant is also changed. Two infected persons are not guaranteed to become resistant, only those who met with resistant are possible to change. Formula (1) explains the process in detail. In the formula, *S* means the number of *susceptible* at time *t*. *I* means the number of *infected* at time *t*, *R* means the number of *resistant* at time *t*, while α represents the infection rate or the infected capability of the *infected*. β represents the resistant capability from which a *infected* would turn into *resistant*. γ represents the possibility that a *susceptible* would turn into *resistant*.

$$\begin{cases} \frac{dS}{dt} = -\alpha(t)SI - \gamma(t)SR \\ \frac{dI}{dt} = \alpha(t)SI - \beta(t)IR \\ \frac{dR}{dt} = \beta(t)IR + \gamma(t)SR \end{cases} \tag{1}$$

3.2 Numerical Simulation

According to the formula (1), we simulate the process of rumor spread by using the SISR model. From numerical simulation, we can identify three different states of SISR model, as Figs. 2, 3, 4 show. The first state is the rumor spread at the widest range when α is large and β is small. At this time, the value of γ matters not much in the result.

Figure 3 shows another more common state of SISR model. α and β hold similar values while the value of γ is relatively high. This state shows that the rumor climbs to a certain height and drops at a speed relevant to γ.

Figure 4 shows that when γ is too small to defeat the rumor. Two different messages (rumor and non-rumor) coexist at the final stage. From the numerical simulation,

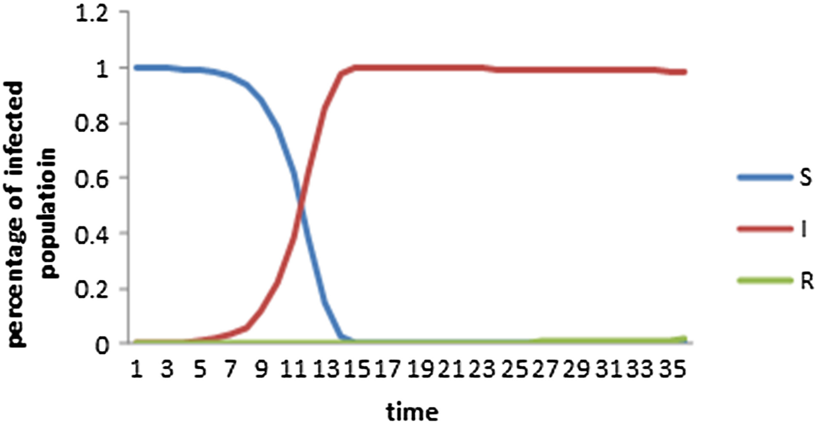


Fig. 2. State of which most population are infective at a short time

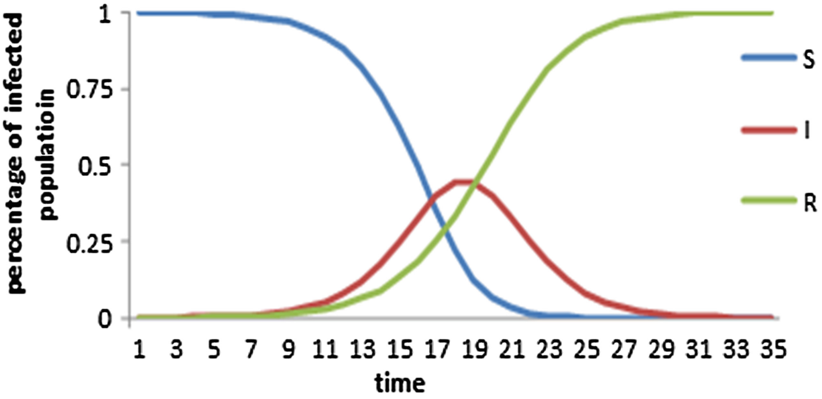


Fig. 3. State of which the rumor starts to evolve and decline after an anti-rumor message is announced

we conclude that when the rumor is more infective, the more population is likely to get involved. When the rumor is being clarified by the authority, the acceptability of the anti-rumor message is the key factor to suppress the rumor.

By exploring the trends, we can easily conclude that as long as the anti-rumor message is accepted, the rumor would finally die out.

4 Experiments

4.1 Real-Life Rumor Simulation

In this section, we would examine some real-life rumors from which we aim to explore the underlying features of online rumors. All data are crawled from Sina Weibo and the

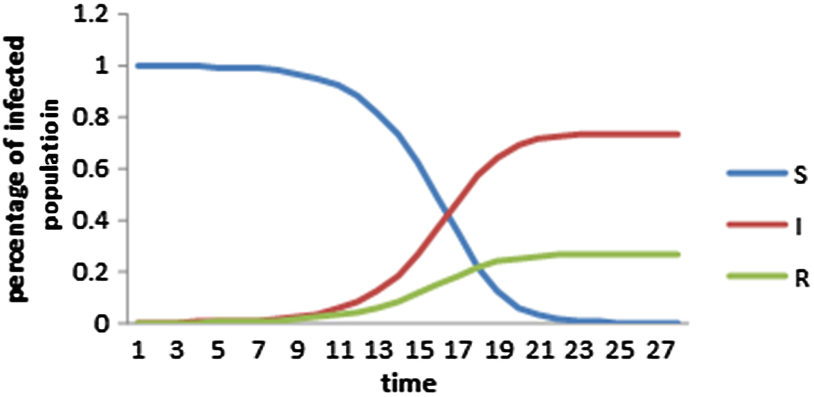


Fig. 4. State of which rumor and anti-rumor message coexists, which means the anti-rumor is not accepted by those infected

incidents have proved to be rumors. In real-world dataset, we obtain the data for S, I and R by using the "timestamp" and "retweet" filed. Firstly, we divide the process of rumor spreading into several spans, the most time several days. At beginning, all the netizens in the network are the *susceptible*. In each span, we define that the netizens who retweet the rumor are *infected*. The netizens who retweet the rumor in last span and do not retweet the rumor again are *resistant*.

Figure 5 is the degree distribution of Liuxiao Lingtong event. The underlying topology of the rumor spread possesses the properties of the scale-free network with a parameter of 2.58. The properties of the scale-free network imply the rumor spread is quite similar to normal information spread. Yet there are still differences between them.

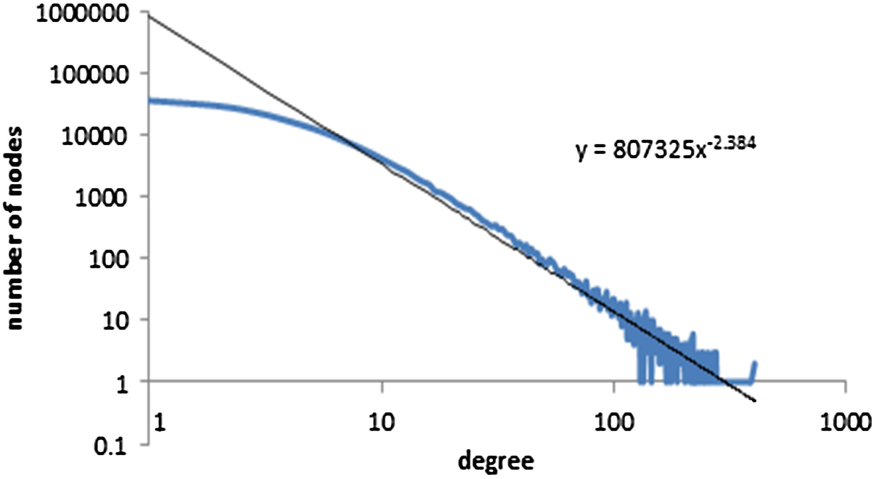


Fig. 5. The degree distribution from an real-life online rumor

Figure 6 represents the stage transition of Liuxiao Lingtong event. From the picture, we can find out that rumors, compared to other messages, would rely much more on the outer relationship. It means that rumors have to absorb new susceptible when they are spread out.

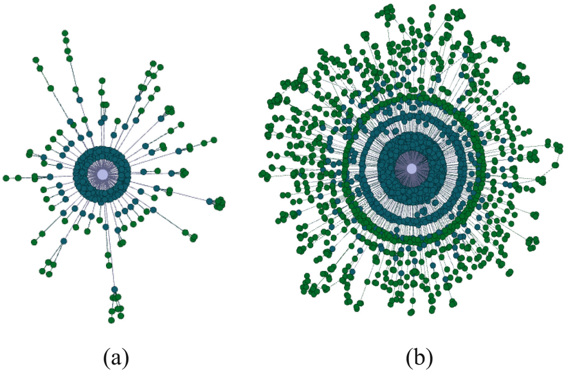


Fig. 6. The change of topology of Liuxiao Lingtong event: (a) stands for the initial state, (b) stands for the middle state, (c) stands for the final state

We use SISR model to fit the real life online rumor data, as Fig. 7 shows. From the picture we can see that the SISR model does a good job when it comes to simulate the real rumor data. Moreover, we can see that the rumor starts to decline when the anti-rumor messages have been announced. Such change could also be reflected in the SISR model. Parameters achieved in experiment for different rumor spread are showed in Table 1.

The rumors, compared to classic rumor spread model, strike out the role of anti-rumor power by adding the shortcut from *S* to *R*. Also, we modified the transition condition of MT model, removing the constraint that *I* meets another *I* would result in the possible occurrence of *R*. As shown in Fig. 8, SISR model fits the real data better than MT model.

4.2 SISR Model on Scale-Free Network

In this section, we examine the characteristics of SISR model on scale-free network. Previous researches showed that no matter where the source, the final stage of the infected population was similar. Yet we contend that there are some people that are more important than the others. The evaluator we use here is k-shell decomposition, which is more accurate in depicting the importance of a node than degree. A vertice with K-shell value k has a degree of at least k with other vertices in the network. The concept of a k-shell was introduced to study the clustering structure of social networks and to describe the evolution of random graphs [14]. We use Ks to donate the K-Shell value of a source node of the rumor and Kt represents the K-shell value of the corresponding source of anti-rumor. Figure 9 is the proportion of k-shell at threshold 10.

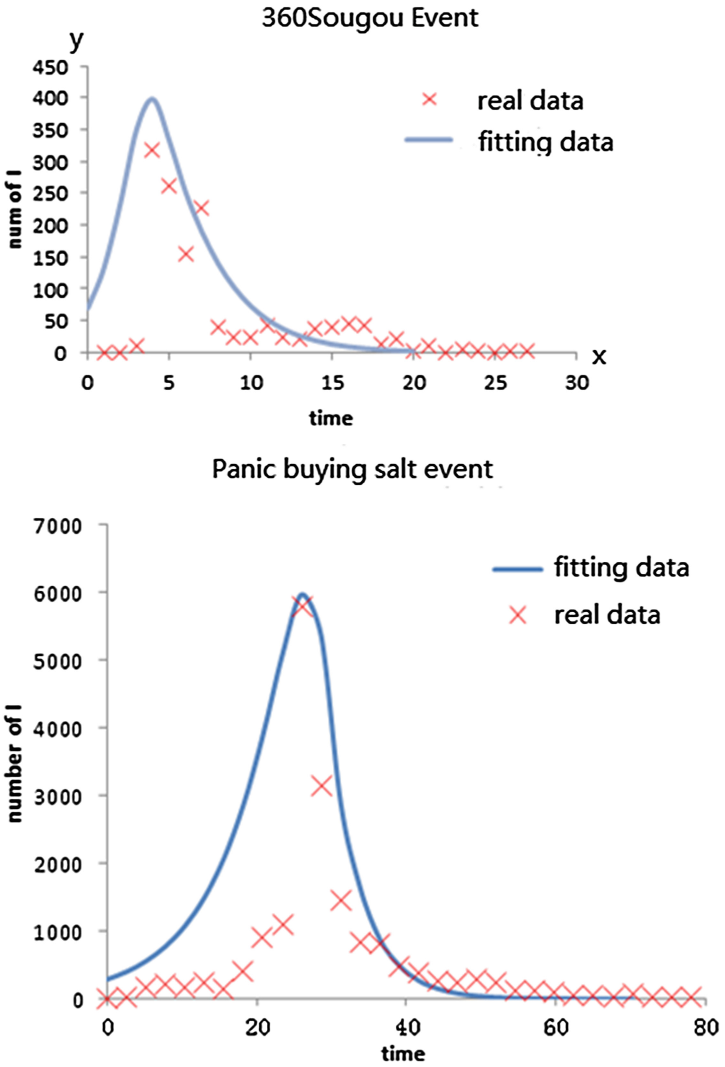


Fig. 7. (a) The fitting data of 360 Sougou event (b) The fitting data of Panic buying salt event

Table 1. Parameters in experiment of some rumor spread cases with SISR model

Rumor	α	β	γ
Liuxiao Lingtong	0.0000092	0.000013	0.000004
Panic buying Salt	0.0000013	0.000006	0.0000015
360 Sougou Event	0.0007	0.0008	0.00028
Big Yellow Duck	0.00004	0.00007	0.00001

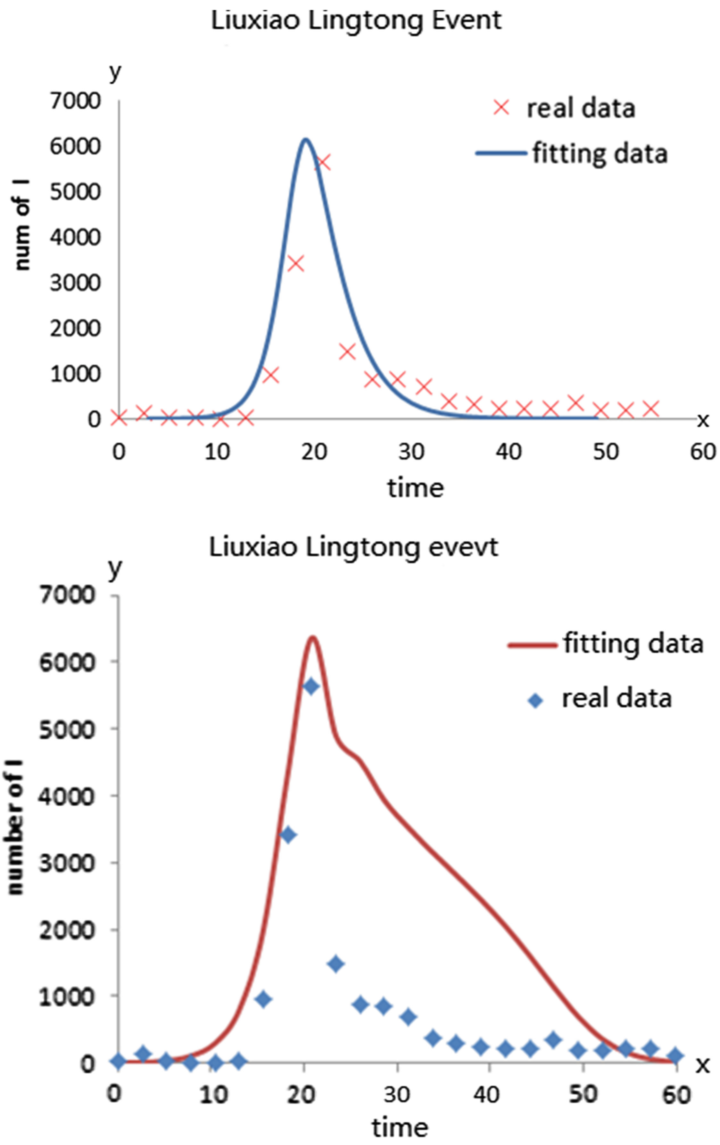


Fig. 8. (a) The fitting data of Liuxiao Lingtong event by SISR model (b) The fitting data of Liuxiao Lingtong event by MT model

We can find out that the percentage of important nodes are nearly 20 % of the whole population. According to 80–20 rule, we assume the important people hold the 20 % value.

We are interested in when a node located near the core of the network heard the rumor. As Fig. 10 shows, nodes live more closely to the core tends to hear the rumor

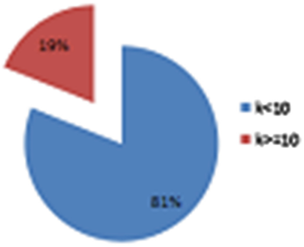


Fig. 9. The proportion of k-shell at threshold 10

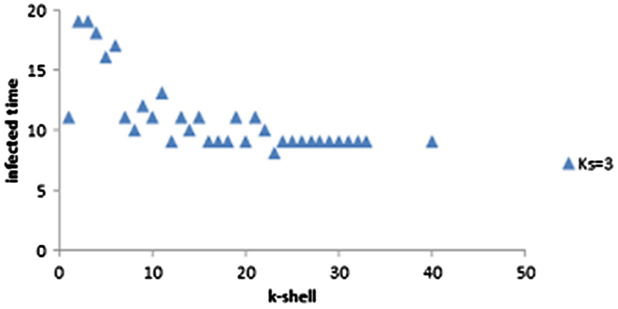


Fig. 10. Nodes of higher k-shell receive the rumor at an earlier time

first compared to the peripheral nodes. Thus we make the assumption that by monitoring nodes in the core level, the authority can easily detect the occurrence of rumors.

We use the maximum infected time to record the time that the last node is infected in network. Figure 11 shows the maximum infected time in the case that an anti-tumor takes action when it detects a rumor spreading in the process. The red line donates that the anti-rumor originated from a node with high k-shell value while the rumor originated from a lower one. Results show that the rumor would die out fast if the detect

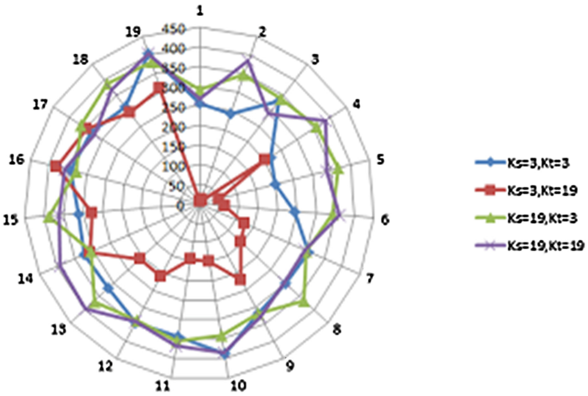


Fig. 11. The maximum infected time (Color figure online)

time is short enough. Yet when the rumor and anti-rumor locate in a similar position in the network, rumors tend to stay a longer time.

The maximum infected size is used to describe the total number of infected nodes in the process of rumor spread in the network. Figure 12 shows the maximum infected size when rumor started from different sources located in the network topology. The red line shows that the anti-rumor is more influential in the network and the anti-rumor message is released earlier, the less people are infected by the rumor.

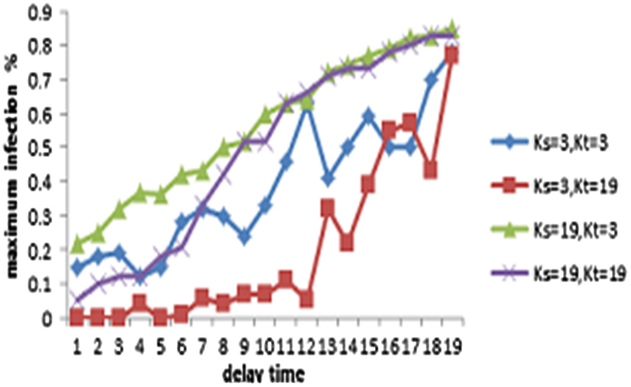


Fig. 12. The infected peak size of a rumor

5 Conclusion and Future Work

In this paper, we proposed SISR model to better simulate the process of rumor spreading. SISR model assumes that instead of turning into a silent stage, those who heard the rumor would try to refute the misinformation. The resistant played an important role in rumor spreading process, in that they help smother the rumor by broadcasting anti-rumor information. The fact that rumors finally die out is the result of the presence of anti-rumor information. SISR model takes rumor resistants into account and quantifies their effect using a set of differentiate equations. Compared with the classic MK model, the revised SISR model depicts the rumor spreading trends in a better way. The SISR model could reflect the lifecycle of a rumor, from getting momentum to dying out. Besides that, we use k-shell decomposition to evaluate the importance of a node transmitting the rumor or anti-rumor spreading process. The larger k-shell value means the node resides in a more important position. The messages sent from a key node reach further than messages sent from a plain node. The more convincing the anti-rumor messages are, the shorter the life of the rumor is. Also, experiments results show that the timing of the appearance of the anti-rumor greatly influence the rumor life. If the anti-rumor comes out earlier, the sooner the rumor dies out.

Extensive data are needed in order to verify SISR model. Currently we just consider cases of single data sources. In the future, multiple data sources, like micro-blogging system, BBS and news portals, could be taken into account and their interactions could be studied. Last but not least, data mining techniques could be used to find out the features of a rumor and how they affects the parameters of the SISR model which basically determines the spreading process and the result of the rumors.

References

1. Daley, D.J., Kendall, D.G.: Epidemics and rumours. Nat. Sci. **204**, 1464–3634 (1964)
2. Maki, D., Thompson, M.: Mathematical Models and Applications: With Emphasis on the Social, Life, and Management Sciences. Prentice- Hall, NJ (1973)
3. Nekovee, M., Moreno, Y., Bianconi, G., Marsili, M.: Theory of rumor spreading in complex social networks. Phy. A **374**(1), 457–470 (2007)
4. Zanette, D.: Critical behavior of propagation on small-world networks. Phys. Rev. E **64**(4), 050901 (2001)
5. Moreno, Y., Nekovee, M., Pacheco, A.: Dynamics of rumor spreading in complex networks. Phys. Rev. E **69**, 066130 (2004)
6. Thompson, R., Estrada, R.C., Daugherty, D., Clintron-Arias, A.: A deterministic approach to the spread of rumors. Technical report, Mathematical and Theoretical Biology (2003)
7. http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201207/t20120723_32497.htm
8. Feng, P., Wang, X., Li, X.: A simulation research on rumors spread based on variable clustering coefficient scale-free networks. J. Syst. Simul. **18**(8), 2346–2348 (2006)
9. Singh, A., Singh, Y.N.: Rumor spreading and inoculation of nodes in complex networks. In: Proceedings of the 21st International Conference Companion on World Wide Web. ACM (2012)
10. Zanette, D., Kuperman, M.: Effects of immunization in small-world epidemics. Phy. A **309** (3), 445–452 (2002)
11. Budak, C., Agrawal, D., El Abbadi, A.: Limiting the spread of misinformation in social networks. In: Proceedings of the 20th International Conference on World Wide Web, WWW ’11, Hyderabad, India, pp. 665–674. ACM (2011)
12. Tripathy, R.M., Bagchi, A., Mehta, S.: A study of rumor control strategies on social networks. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM ’10, Toronto, ON, Canada, pp. 1817–1820. ACM (2010)
13. Kermack, W.O., McKendrick, A.G.: A contribution to the mathematical theory of epidemics. Proc. Roy. Soc.: Math. Phys. Eng. Sci. **115**(772), 700–721 (1927)
14. Foster, B.L., Seidman, S.B.: A formal unification of anthropological kinship and social network methods. In: Research Methods in Social Network Analysis, pp. 41–59 (1989)

GS4: Generating Synthetic Samples for Semi-Supervised Nearest Neighbor Classification

Panagiotis Moutafis(✉) and Ioannis A. Kakadiaris

Computational Biomedicine Lab, Department of Computer Science,
University of Houston, Houston, TX 77004-2693, USA
{pmoutafis,ioannisk}@uh.edu
http://www.cbl.uh.edu/

Abstract. In this paper, we propose a method to improve nearest neighbor classification accuracy under a semi-supervised setting. We call our approach GS4 (i.e., Generating Synthetic Samples Semi-Supervised). Existing self-training approaches classify unlabeled samples by exploiting local information. These samples are then incorporated into the training set of labeled data. However, errors are propagated and misclassifications at an early stage severely degrade the classification accuracy. To address this problem, the proposed method exploits the unlabeled data by using weights proportional to the classification confidence to generate synthetic samples. Specifically, our scheme is inspired by the Synthetic Minority Over-Sampling Technique. That is, each unlabeled sample is used to generate as many labeled samples as the number of classes represented by its k-nearest neighbors. In particular, the distance of each synthetic sample from its k-nearest neighbors of the same class is proportional to the classification confidence. As a result, the robustness to misclassification errors is increased and better accuracy is achieved. Experimental results using publicly available datasets demonstrate that statistically significant improvements are obtained when the proposed approach is employed.

Keywords: K-nearest neighbor · Semi-supervised learning · Synthetic samples · Classification

1 Introduction

The k-nearest neighbors (kNN) rule [5] is one of the most popular classifiers due to its numerous advantages (e.g., simplicity and mild structural assumptions [8]). Therefore, enhancing its accuracy will benefit numerous applications. In particular, the kNN rule classifies each testing sample using the majority labels of its k-nearest neighbors in the training set. As a result, its performance is significantly affected by the size of the training set. Specifically, the accuracy of kNN significantly drops when the number of training samples is small [13].

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 393–403, 2014.
DOI: 10.1007/978-3-319-13186-3_36

Several approaches seek to exploit information from unlabeled samples to address this problem. These methods fall within the domains of transductive learning [6] and semi-supervised learning [2]. The former exploits information from the set of test samples to increase the accuracy on the test set itself, while the latter exploits information from a set of unlabeled samples to enhance the accuracy on a separate test set.

There are four types of semi-supervised learning methods: (i) generative models, (ii) multi-view, (iii) self-training, and (iv) graph-based. Generative models make assumptions concerning the distribution of the data and incorporate the unlabeled samples accordingly. However, improved accuracy is guaranteed only when the assumed generative model is correct which is not usually the case [4]. Multi-view algorithms co-train two classifiers using different features or different training sets. Then, their most confident predictions of the unlabeled data are used to incorporate samples to each other's training sets. However, there may not always be a natural split of the features. In addition, splitting the training set for kNN classification would not result in increased accuracy. The reason is that kNN uses local relationships in the training set to classify test samples. Hence, splitting the training set would not produce kNN classifiers with the ability to co-train each other. Self-training methods iteratively incorporate into the training set unlabeled samples for which the classification confidence is high. Such techniques are easy to implement but they propagate errors which may severely degrade the accuracy. Finally, graph-based approaches consider the shape of the data to determine the classification confidence. However, estimating the graph is computationally expensive. In addition, a good solution may be found only if the graph is a good representative of the data distribution.

Most of the semi-supervised approaches in the context of kNN classification use label propagation to gradually incorporate unlabeled samples to the training set. Some methods rely exclusively on local consistency. For example, the Propagating 1-Nearest-Neighbor (Prop-1) method [13] is a self-training technique that relies on 1NN classification. On each iteration the unlabeled sample that has the minimum distance to its nearest neighbor is incorporated to the training set. Several other approaches have been proposed that take into consideration global features using graph-based or sub-manifold representation of the data. Two such examples are the Learning from Labeled and Unlabeled Data with Label Propagation [12] and Learning with Local and Global Consistency [11]. However, methods that use label propagation assign each unlabeled sample to only one class. Consequently, the classification boundary is changing abruptly in each iteration. Moreover, errors are propagated which may result in many misclassifications. To address this problem, a statistical approach was proposed by Ghosh [7] that considers all the possible choices of labels for the test samples and then aggregates them by using the Bayesian model averaging technique.

In this paper, we propose a self-training method that addresses the problem of assigning each unlabeled sample to a single class. To this end, each unlabeled sample is first ranked in terms of classification confidence according to two criteria: (i) class distribution of its k-nearest neighbors, and (ii) average distance

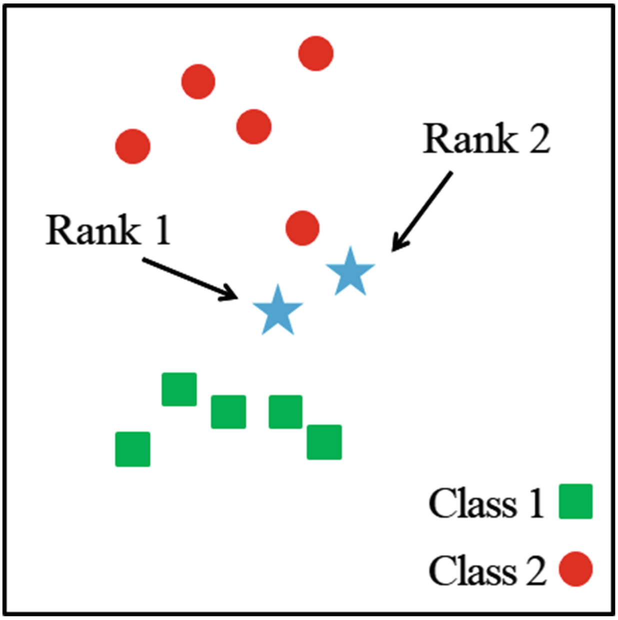


Fig. 1. Overview of the proposed ranking scheme. The points indicated by a star represent the unlabeled samples.

from its k-nearest neighbors, computed for each of the classes independently (see Fig. 1). The rank-1 unlabeled sample is then used to generate as many synthetic samples as the number of classes of its k-nearest neighbors following a procedure similar to the Synthetic Minority Over-Sampling Technique [3]. In particular, a new sample is generated for each class and it is positioned between the rank-1 unlabeled sample and the median of the training samples of the corresponding class (see Fig. 2). Note that the median for each class is computed using samples only from the k-nearest neighbors. Other measures of central tendency may be used as well (e.g., mean operator). However, the median is robust to outliers and we observed that it yields better results. The higher the confidence that the unlabeled sample belongs to a class the closer the synthetic sample is positioned to the unlabeled sample. If the confidence is low then the synthetic sample is positioned closer to the median of the training samples that belong to that class. To this end, we propose a confidence measure based on the weighted sum of the estimated conditional class probability and the normalized average distance of the k-nearest neighbors of that class. The synthetic samples are incorporated into the training set, the unlabeled sample is discarded, and the whole process is repeated. The benefits of the proposed method are: (i) increased robustness to misclassifications, and (ii) the alteration of the decision boundary is proportional to the classification confidence for each class. Furthermore, the ranking

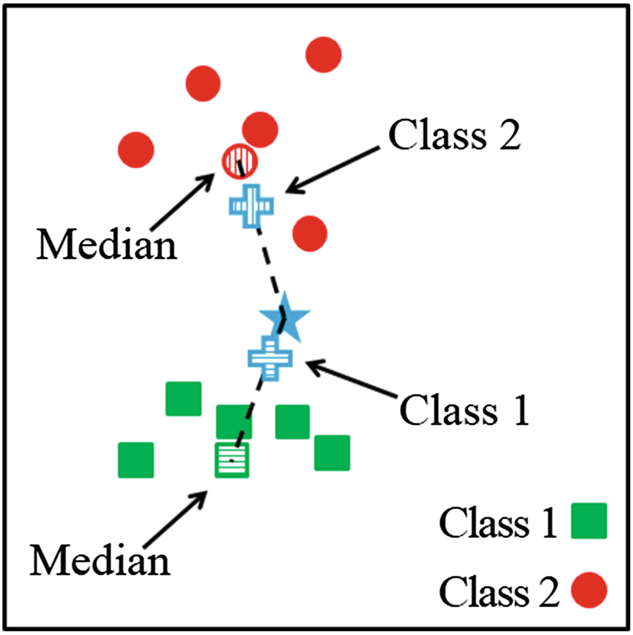


Fig. 2. Overview of the generation of synthetic samples steps. The points indicated by a cross represent the synthetic samples generated.

scheme presented in this paper can be used to extend Prop-1 to k neighbors. An overview of the proposed method is presented in Figs. 1 and 2.

The rest of the paper is organized as follows: in Sect. 2 the proposed approach and the corresponding algorithms are described, in Sect. 3 the experimental evaluation is presented, and in Sect. 4 our conclusions are discussed.

2 Method

In this paper, we focus on a transductive setting due to the simplicity that it provides in describing and evaluating the proposed method. Given a training set $\mathbf{T} = \{(x_i^t, l_i^t) | x_i^t \epsilon \mathbb{R}^n, l_i^t \epsilon \mathbf{C},$ and $i = 1, \cdots, m_t\}$, and a validation set $\mathbf{V} = \{x_i^v | x_i^v \epsilon \mathbb{R}^n$ and $i = 1, \cdots, m_v\}$, where x_i^t is a training sample, n is the number of features, l_i^t is the corresponding label, $\mathbf{C} = \{c_1, \cdots, c_z\}$ is the set of labels, m_t is the number of training samples, x_i^v is a validation sample, and m_v is the number of validation samples. The objective of transductive learning is to exploit the information of the validation set $\mathbf{V}$ to increase the accuracy on $\mathbf{V}$ itself. Throughout this paper, capital bold font is used to denote sets. We divide the proposed method GS4 (i.e., Generating Synthetic Samples Semi-Supervised) into two parts: (i) selection of unlabeled sample, and (ii) generation of synthetic samples.

2.1 Selection of Unlabeled Sample

To select the unlabeled sample to be used for the generation of synthetic samples all of the unlabeled samples are first ranked based on their classification confidence. In particular, the first criterion relies on the posterior probabilities $P(l_j^v = c_r|, x_j^v \mathbf{T}, k)$:

$$P(l_j^v = c_r|x_j^v, \mathbf{T}, k) = \sum_{x_i^t \epsilon \mathcal{N}_k(x_j^v)} \frac{I(c_r = l_j^v)}{k}, \tag{1}$$

where $\mathcal{N}_k(x_j^v)$ is a closed ball (neighborhood) such that it includes the k-nearest neighbors x_i^t of x_j^v, and $I(c_r = l_j^v)$ is an indicator function that equals 1 when $c_r = l_j^v$, and 0 otherwise. The higher the value of this probability for a given unlabeled sample x_j^v the higher the confidence that its class label is c_r. However, using this criterion to rank the unlabeled samples according to their classification confidence would not be very effective because it usually yields many ties. This effect is more pronounced when the number of neighbors is small, which is usually the case for semi-supervised and transductive learning. To address this problem, a second criterion is employed based on the average distance of x_j^v to the samples of the class c_r that belong in $\mathcal{N}_k(x_j^v)$:

$$A(x_j^v, l_i^t = c_r|\mathbf{T}, k) = \sum_{x_i^t \epsilon \mathcal{N}_k(x_j^v)} \|(x_j^v - x_i^t) \cdot I(c_r = l_i^t)\|_2^2 \Big/ \sum_{x_i^t \epsilon \mathcal{N}_k(x_j^v)} I(c_r = l_i^t). \tag{2}$$

If $\sum_{x_i^t \epsilon \mathcal{N}_k(x_j^v)} I(c_r = l_i^t) = 0$ we set $A(x_j^v, l_i^t = c_r|\mathbf{T}, k) = 0$. The smaller the average distance $A(x_j^v, l_i^t = c_r|\mathbf{T}, k)$ the higher the confidence that x_j^v is labeled as c_r. As implied, $A(x_j^v, l_i^t = c_r|\mathbf{T}, k)$ is used to break the ties obtained by $P(l_j^v = c_r|x_j^v, \mathbf{T}, k)$. To this end, the $A(x_j^v, l_i^t = c_r|\mathbf{T}, k)$ values are summed together with $P(l_j^v = c_r|x_j^v, \mathbf{T}, k)$. However, in order not to alter the ranking between the different values of $P(l_j^v = c_r|x_j^v, \mathbf{T}, k)$ the range of $A(x_j^v, l_i^t = c_r|\mathbf{T}, k)$ needs to be scaled accordingly. In practice, it is enough to multiply all the values obtained for $A(x_j^v, l_i^t = c_r|\mathbf{T}, k)$ with a small quantity ε. The confidence of each unlabeled sample x_j^v is then determined based on the class label c_r that maximizes the sum of the two terms as follows:

$$F(x_j^v|\mathbf{T}, k) = max_{c_r}\Big[P(l_j^v = c_r|x_j^v, \mathbf{T}, k) + \varepsilon \cdot A(x_j^v, l_i^t = c_r|\mathbf{T}, k)\Big]. \tag{3}$$

Finally, the set $\mathbf{R}$ is computed that comprises of all the 2-tuples of the form $(j, rank)$. The rank for each index j is computed based on the values produced by $F(x_j^v|\mathbf{T}, k)$, $\forall j$. Any remaining ties are broken randomly. An overview of the ranking scheme is provided by Algorithm 1.

2.2 Generation of Synthetic Samples

The synthetic samples are generated each time for the rank-1 unlabeled sample. For each sample, the average distances $A(x_j^v, l_i^t = c_r|\mathbf{T}, k)$ are normalized across

Algorithm 1. GS4: Selection of Unlabeled Sample

```
Input: $\mathbf{T}$, $\mathbf{V}$, and $k$
Output: $\mathbf{R}$
for $j = 1, \ldots, m_v$ do
    Compute $P(l_j^v = c_r | x_j^v, \mathbf{T}, k),\ \forall c_r \epsilon \mathcal{C}$ according to Eq. (1).
    Compute $A(x_j^v, l_i^t = c_r | \mathbf{T}, k),\ \forall c_r \epsilon \mathcal{C}$ according to Eq. (2).
end for
Compute $F(x_j^v | \mathbf{T}, k),\ \forall j$ according to Eq. (3).
Compute $\mathbf{R}$ using $F(x_j^v | \mathbf{T}, k),\ \forall j$.
```

Algorithm 2. GS4: Generation of Synthetic Samples

```
Input: $\mathbf{T}$, $\mathbf{V}$ and $k$
Output: $\mathbf{T}$
for $j = 1, \ldots, m_v$ do
    Compute $\mathbf{R}$ using Alg. 1.
    for $r = 1, \ldots, z$ do
        Generate $s_r = G(x^v_{\mathbf{R}_{(1)}}, \mathbf{R}_{(1)}, r | \mathbf{T}, k)$ according to Eq. (4).
    end for
    Update $\mathbf{T} = \mathbf{T} \cup \{s_r\}\ \forall r$.
    Update $\mathbf{V} = \mathbf{V} \cap \{x^v_{\mathbf{R}_{(1)}}\}^c$.
end for
```

the class labels in such a way that they sum to a unit (i.e., $\sum_{c_r} A(x_j^v, l_i^t = c_r | \mathbf{T}, k) = 1$). The normalized average distances are denoted by $A^*(x_j^v, l_i^t = c_r | \mathbf{T}, k)$. Then, the overall confidence level is computed as a weighted sum of $A^*(x_j^v, l_i^t = c_r |, \mathbf{T}, k)$ and $P(l_j^v = c_r | x_j^v, \mathbf{T}, k)$. Specifically, the synthetic sample for each class label c_r is computed as:

$$\begin{aligned} G(x_j^v, j, r | \mathbf{T}, k) =& median(\{x_i^t | x_i^t \epsilon \mathcal{N}_k(x_j^v) \cap l_i^t = c_r\}) \\ &+ \frac{1}{2}\Big(A^*(x_j^v, l_i^t = c_r | \mathbf{T}, k) + P(l_j^v = c_r | x_j^v, \mathbf{T}, k)\Big) \\ &\cdot \Big(x_j^v - median(\{x_i^t | x_i^t \epsilon \mathcal{N}_k(x_j^v) \cap l_i^t = c_r\})\Big). \end{aligned} \tag{4}$$

The geometric interpretation of Eq. (4) is illustrated in Fig. 2. Note that when the confidence is high the synthetic sample is almost equal to the unlabeled sample. Hence, the classification boundary is extended farther. On the other hand, when the confidence is low the synthetic sample is closer to the median of the training samples within $\mathcal{N}_k(x_j^v)$ labeled as c_r. Consequently, the classification boundary is less affected. The median operator was preferred due to robustness to outliers. An overview of the synthetic samples generation algorithm is provided in Algorithm 2, where $\mathbf{R}_{(1)}$ denotes the index j of the rank-1 unlabeled sample.

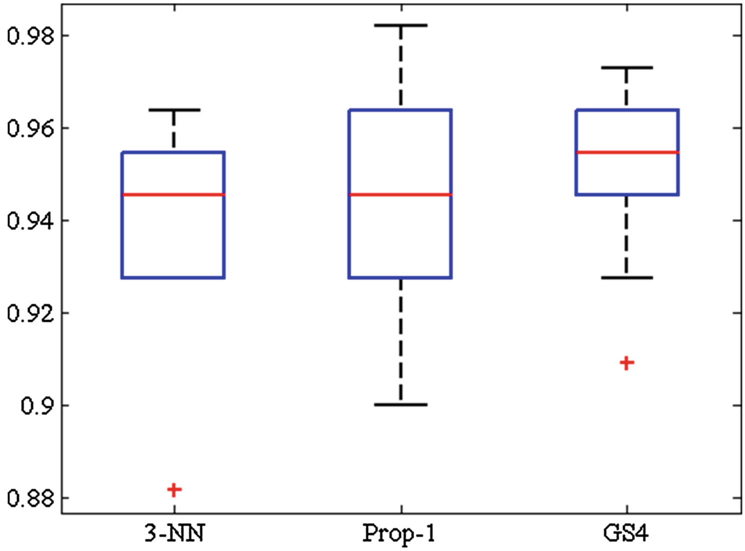


Fig. 3. Overview of results for the Iris database. Depicted are the boxplots of the baselines 3-NN (i.e., 3-Nearest Neighbor Classification) and Prop-1 (i.e., Propagating 1-Nearest-Neighbor) [13], and the proposed approach GS4 (i.e., Generating Synthetic Samples Semi-Supervised).

3 Experimental Evaluation

To assess the effectiveness of GS4 we conducted experiments using four datasets from the University of California repository [9]. Specifically, we used the Iris, Wine, Balance, and Breast Diagnostic (BD) datasets. These datasets are well known in the research community and offer a variability in terms of number of samples, features, classes, and difficulty. A brief overview of the datasets is provided in Table 1. We used 20 %/80 % splits for training and validation under a transductive setting. We repeated this procedure 100 times and computed the corresponding mean and standard deviation values of the classification accuracy. As a baseline we used our own implementation of Prop-1 because this is the only

Table 1. Overview of the datasets used. Note that z denotes the number of classes, n the number of input features, and $m = m_t + m_v$ the total number of samples.

Dataset	Attributes	z	n	m
Iris	Continuous	3	4	150
Wine	Continuous	3	13	178
Balance	Categorical	3	4	625
BD	Continuous	2	30	569

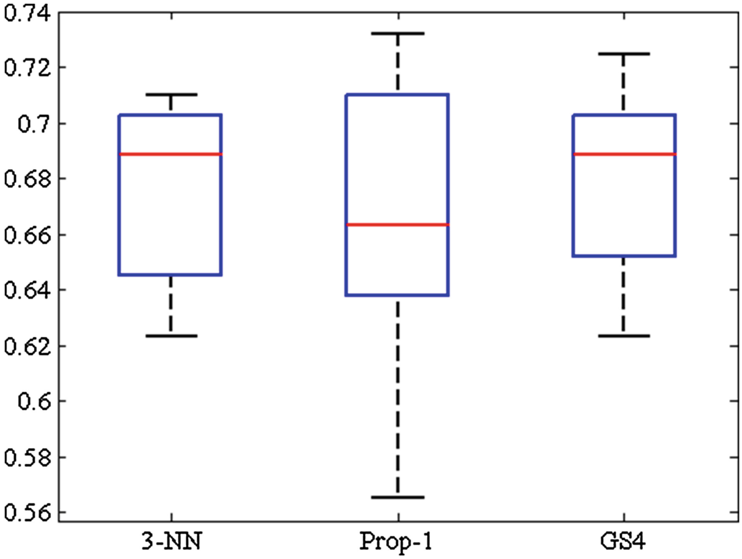


Fig. 4. Overview of results for the Wine database. Depicted are the boxplots of the baselines 3-NN (i.e., 3-Nearest Neighbor Classification) and Prop-1 (i.e., Propagating 1-Nearest-Neighbor) [13], and the proposed approach GS4 (i.e., Generating Synthetic Samples Semi-Supervised).

Table 2. Overview of the obtained results. The values refer to accuracy in percentages in the form *mean (standard deviation)*. Bold font is used to denote the best performance. The *p-values* for the median correspond to one-sided non-parametric Wilcoxon Signed-Rank tests. The *p-values* for the variance correspond to non-parametric Brown-Forsythe tests. The individual statistical significance level after the Bonferonni adjustment is 0.63 %. Bold font is used to denote statistically significant improvements.

Dataset	3-NN	Prop-1	GS4	Median	Variance
Iris	94.1 (2.3)	94.4 (2.5)	**95.1** (2.0)	$\mathbf{8.2 \cdot 10^{-5}}$	$\mathbf{1.9 \cdot 10^{-3}}$
Wine	67.6 (3.2)	66.4 (4.9)	**68.0** (3.1)	$\mathbf{2.6 \cdot 10^{-4}}$	$\mathbf{2.6 \cdot 10^{-5}}$
Balance	81.1 (1.7)	79.7 (0.0)	**84.2** (0.2)	$\mathbf{1.7 \cdot 10^{-18}}$	$\mathbf{4.8 \cdot 10^{-13}}$
BD	**91.4** (0.0)	91.3 (0.0)	91.3 (0.0)	$2.3 \cdot 10^{-1}$	$2.4 \cdot 10^{-1}$

self-training approach in the context of kNN classification. Note that for both Prop-1 [13] and GS4 only 50 % of the unlabeled data were exploited to enhance the training set. Our rationale behind this decision is that the bottom 50 % of the data in terms of classification confidence would add noise and degrade the overall accuracy. Moreover, we computed the accuracy obtained under the corresponding supervised setting (3-NN). In all cases, the number of neighbors k was set to three. To assess the statistical significance of the obtained results we performed one-sided non-parametric Wilcoxon Signed-Rank tests [10]. Specifically, the null hypothesis was set to H_0: *the Prop-1 and GS4 median accuracies are* *equal*, and the alternative hypothesis was defined as H_a: *the GS4 median accuracy is higher compared to the Prop-1 median accuracy.* Moreover, we performed non-parametric Brown-Forsythe tests [1]. In particular, the null hypothesis was defined as H_0: *the variances of the GS4 and Prop-1 accuracy values are equal*, and the alternative hypothesis was set to H_a: *the variance of the GS4 accuracy values is different than the variance of the Prop-1 accuracy values.* The Bonferonni adjustment was used to ensure that the overall statistical significance remains 5 % due to the multiple tests performed. That is, the statistical significance of each individual test was set to $\frac{5\,\%}{8} = 0.63\,\%$. A summary of the results along with the p-values of the tests performed is presented in Table 2. The corresponding boxplots are depicted in Figs. 3, 4, 5 and 6. Based on these results, it appears that the accuracy of GS4 is statistically significantly higher than the accuracy of Prop-1 for the Iris, Wine, and Balance datasets. In addition, the corresponding standard deviations are statistically significantly lower. For the BD dataset both methods degrade the baseline performance. However, GS4 results in smaller accuracy degradation compared to that of Prop-1 (i.e., 91.31 % and 91.25 %, respectively). This provides further evidence that GS4 is more robust than Prop-1.

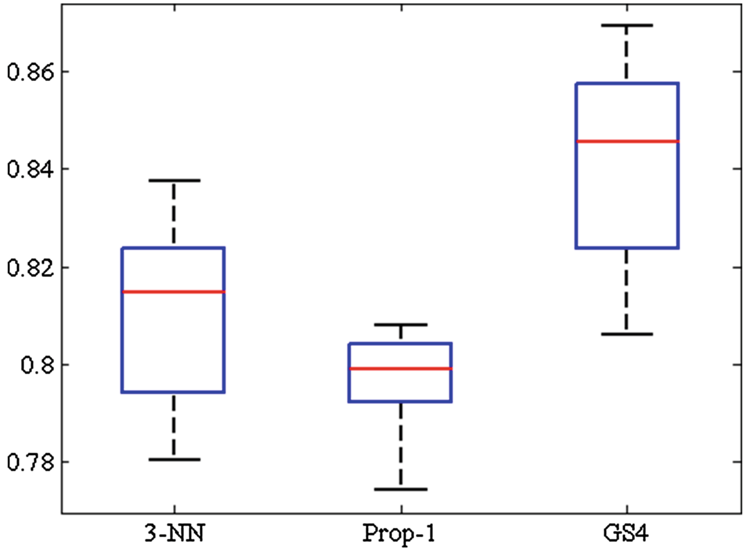


Fig. 5. Overview of results for the Balance database. Depicted are the boxplots of the baselines 3-NN (i.e., 3-Nearest Neighbor Classification) and Prop-1 (i.e., Propagating 1-Nearest-Neighbor) [13], and the proposed approach GS4 (i.e., Generating Synthetic Samples Semi-Supervised).

Finally, the training time for GS4 is higher compared to Prop-1 due to the additional operations performed and the generation of synthetic samples. However, prototyping methods can be employed on the enhanced training set to reduce the time complexity during testing and further increase its robustness.

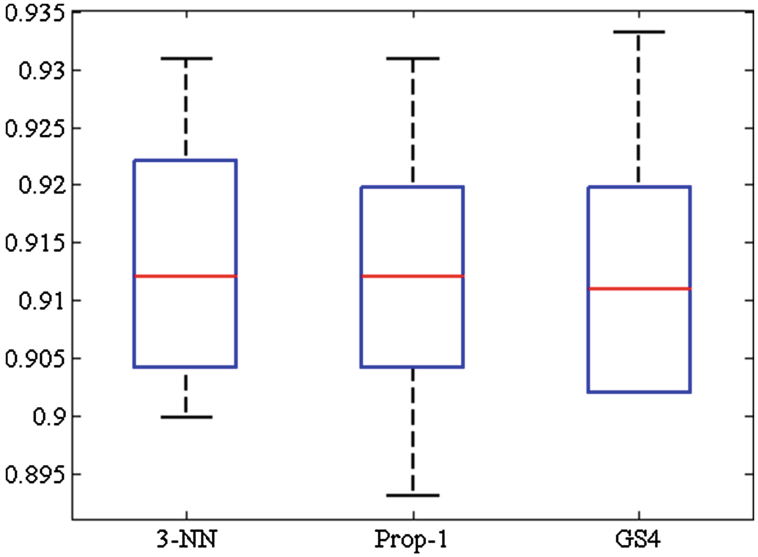


Fig. 6. Overview of results for the Breast Diagnostic database. Depicted are the boxplots of the baselines 3-NN (i.e., 3-Nearest Neighbor Classification) and Prop-1 (i.e., Propagating 1-Nearest-Neighbor) [13], and the proposed approach GS4 (i.e., Generating Synthetic Samples Semi-Supervised).

4 Conclusion

In this paper, we introduced a method that exploits unlabeled data to generate synthetic samples with the goal of increasing the kNN accuracy. We demonstrated that the proposed method is more robust compared to existing self-training approaches. The experimental results indicate that statistically significant improvements in terms of median accuracy and variance were obtained for GS4 over Prop-1 in three of the four datasets used, and a better accuracy for the last one.

Acknowledgments. This research was funded in part by the US Army Research Lab (W911NF-13-1-0127) and the UH Hugh Roy and Lillie Cranz Cullen Endowment Fund. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of the sponsors.

References

1. Brown, M., Forsythe, A.: Robust tests for the equality of variances. J. Am. Stat. Assoc. **69**(346), 364–367 (1974)
2. Chapelle, O., Schölkopf, B., Zien, A.: Semi-supervised Learning, vol. 2. MIT Press, Cambridge (2006)

3. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
4. Cohen, I., Cozman, F., Sebe, N., Cirelo, M., Huang, T.: Semisupervised learning of classifiers: theory, algorithms, and their application to human-computer interaction. IEEE Trans. Pattern Anal. Mach. Intell. **26**(12), 1553–1566 (2004)
5. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theor. **13**(1), 21–27 (1967)
6. Dean, N., Murphy, T., Downey, G.: Using unlabelled data to update classification rules with applications in food authenticity studies. J. Roy. Stat. Soc. Ser. C (Appl. Stat.) **55**(1), 1–14 (2006)
7. Ghosh, A.: A probabilistic approach for semi-supervised nearest neighbor classification. Pattern Recogn. Lett. **33**(9), 1127–1133 (2012)
8. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning Data Mining, Inference and Prediction. Springer, New York (2009)
9. Merz, C., Murphy, P., Aha, D.: UCI repository of machine learning databases. Department of Information and Computer Science, University of California (2012)
10. Wolfe, D., Hollander, M.: Nonparametric Statistical Methods. Wiley Series in Probability and Statistics. Wiley, New York (1973)
11. Zhou, D., Bousquet, O., Lal, T., Weston, J., Schölkopf, B.: Learning with local and global consistency. Adv. Neural Inf. Process. Syst. **16**(16), 321–328 (2004)
12. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Technical report, CMU-CALD-02-107, Carnegie Mellon University (2002)
13. Zhu, X., Goldberg, A.: Introduction to semi-supervised learning. Synth. Lect. Artif. Intell. Mach. Learn. **3**(1), 1–130 (2009)

A Scalable Data Analytics Algorithm for Mining Frequent Patterns from Uncertain Data

Richard Kyle MacKinnon, Carson Kai-Sang Leung(✉), and Syed K. Tanbeer

University of Manitoba, Winnipeg, MB, Canada
kleung@cs.umanitoba.ca

Abstract. With advances in technology, massive amounts of valuable data can be collected and transmitted at high velocity in various scientific, biomedical, and engineering applications. Hence, scalable data analytics tools are in demand for analyzing these data. For example, scalable tools for association analysis help reveal frequently occurring patterns and their relationships, which in turn lead to intelligent decisions. While a majority of existing frequent pattern mining algorithms (e.g., FP-growth) handle only precise data, there are situations in which data are *uncertain*. In recent years, tree-based algorithms for mining uncertain data (e.g., UF-growth, UFP-growth) have been developed. However, tree structures corresponding to these algorithms can be large. Other tree structures for handling uncertain data may achieve compactness at the expense of loose upper bounds on expected supports. In this paper, we propose (i) a compact tree structure that captures uncertain data with tighter upper bounds than aforementioned tree structures and (ii) a scalable data analytics algorithm that mines frequent patterns from our tree structure. Experimental results show the tightness of bounds to expected supports provided by our algorithm.

1 Introduction

As technologies advance, massive amounts of valuable data can be collected and transmitted at high velocity in various scientific, biomedical, and engineering applications. Useful knowledge is embedded in these data. Data mining techniques help analyze these data for the discovery of implicit, previously unknown, and potentially useful knowledge (e.g., clusters [15], selected features [16], graph patterns [6], frequent patterns). Since the advent of frequent pattern mining [1], numerous studies have been conducted to find *frequent patterns* (i.e., *frequent itemsets*) from *precise* data such as databases of shopper market basket transactions [8]. When mining precise data, users definitely know whether an item is present in (or is absent from) a transaction. In this notion, each item in a transaction t_j in databases of precise data can be viewed as an item with a 100 % likelihood of being present in t_j. However, there are situations in which users are uncertain about the presence or absence of items [3,7,10,12]. For example, a meteorologist may suspect (but cannot guarantee) that severe weather phenomena will develop during a thunderstorm. The uncertainty of such suspicions can

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 404–416, 2014.
DOI: 10.1007/978-3-319-13186-3_37

be expressed in terms of *existential probability*. For instance, a thunderstorm may have a 60 % likelihood of generating hail, and only a 15 % likelihood of generating a tornado, regardless of whether or not there is hail.

To mine frequent patterns from uncertain data, the U-Apriori algorithm [4] was proposed in PAKDD 2007. As an Apriori-based algorithm, U-Apriori requires multiple scans of uncertain data. To reduce the number of scans (down to two), the tree-based UF-growth algorithm [11] was proposed in PAKDD 2008. In order to compute the *exact* expected support of each pattern, paths in the corresponding UF-tree are shared only if tree nodes on the paths have the same item and the same existential probability values. The resulting UF-tree may be quite large when compared to the FP-tree [5] (for capturing precise data). In an attempt to make the tree compact, the UFP-growth algorithm [2] groups *similar* nodes (with the same item but similar existential probability values) into a cluster. However, depending on the clustering parameter, the corresponding UFP-tree may be as large as the UF-tree. Moreover, because UFP-growth does not store every existential probability value for an item in a cluster, it returns not only the frequent patterns but also some infrequent patterns (i.e., false positives). As alternatives to trees, hyperlinked array structures were used by the UH-Mine algorithm [2], which was reported [14] to outperform UFP-growth. The PUF-growth algorithm [13] was proposed in PAKDD 2013 to utilize a concept of an upper bound to expected support together with more aggressive path sharing to yield a more compact tree structure, and it was shown to outperform UH-Mine.

In this paper, we study the following questions: Can we further tighten the upper bound on expected support? Can the resulting tree be as compact as the FP-tree? How would frequent patterns be mined from such a tree? Our key contributions of this paper are as follows:

1. the concept of tightened prefixed pattern cap (TPC);
2. a tightened prefix-capped uncertain frequent pattern tree (PUF*-tree) structure, which can be as compact as the original FP-tree while capturing uncertain data; and
3. a scalable data analytics algorithm—called PUF*-growth—which is guaranteed to mine *all and only those* frequent patterns (i.e., *no* false negatives and *no* false positives) from uncertain data.

The remainder of this paper is organized as follows. The next section gives background and related works. Section 3 discusses how to further tighten the upper bound of the expected support. In Sects. 4 and 5, we propose our PUF*-tree structure and PUF*-growth algorithm, respectively. Evaluation results are shown in Sect. 6, and conclusions are presented in Sect. 7.

2 Background and Related Works

Let (i) `Item` be a set of m domain items and (ii) $X = \{x_1, x_2, \ldots, x_k\}$ be a pattern comprising k items (i.e., a *k-itemset*), where $X \subseteq$ `Item` and $1 \leq k \leq m$. Then, each item x_i in a transaction $t_j = \{x_1, x_2, \ldots, x_h\} \subseteq$ `Item` in a transactional database of uncertain data is associated with an *existential probability value* $P(x_i, t_j)$ [9], which represents the likelihood of the presence of x_i in t_j. Note that $0 < P(x_i, t_j) \leq 1$.

The *existential probability* $P(X, t_j)$ of a pattern X in t_j is then the product of the corresponding existential probability values of every item x within X (where these items are independent) [9]: $P(X, t_j) = \prod_{x \in X} P(x, t_j)$. The *expected support* $expSup(X)$ of X in the database of uncertain data is the sum of $P(X, t_j)$ over all n transactions in the database:

$$expSup(X) = \sum_{j=1}^{n} P(X, t_j). \tag{1}$$

Given (i) a database of uncertain data and (ii) a user-specified minimum support threshold *minsup*, the research problem of *frequent pattern mining from uncertain data* is to discover from the database all those *frequent* patterns (i.e., patterns having expected support $\geq$ *minsup*).

Recall from Sect. 1 that the *UF-growth algorithm* [11] uses *UF-trees* to mine frequent patterns from uncertain data in two scans of the data. Each node in a UF-tree captures (i) an item x, (ii) its existential probability, and (iii) its occurrence count. Tree paths are shared if the nodes on these paths share the same item and existential probability. In general, when dealing with uncertain data, it is not uncommon that the existential probability values of the same item vary from one transaction to another. As such, the resulting UF-tree may not be as compact as the FP-tree. Figure 1(a) shows a UF-tree for the uncertain data presented in Table 1 when *minsup* = 1.1. The UF-tree contains four nodes for item a with different probability values as children of the root. Efficiency of the corresponding UF-growth algorithm, which finds all and *only those* frequent patterns, partially relies on the compactness of the UF-tree.

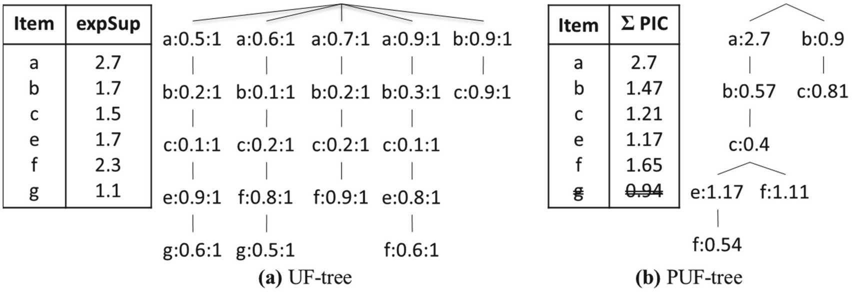
In an attempt to make the tree more compact, the *UFP-growth algorithm* [2] builds a *UFP-tree* (by scanning the uncertain data twice). Tree paths are shared if the nodes on these paths share the same item but *similar* existential probability values. With such a less restrictive path sharing condition, nodes for item x having similar existential probability values are clustered into a mega-node. The resulting mega-node in the UFP-tree captures (i) an item x, (ii) the *maximum* existential probability value (among all nodes within the cluster), and (iii) its occurrence count. By extracting appropriate tree paths and constructing UFP-trees for subsequent projected databases, the UFP-growth algorithm finds all frequent patterns and some *false positives* at the end of the second scan of uncertain data. A third scan is then required to remove those false positives.

As a further attempt to improve the compactness of the tree, the *PUF-growth algorithm* [13] uses a *PUF-tree* to tighten the upper bound on the expected support of patterns. Each node in a PUF-tree captures (i) an item x and (ii) a *prefixed item cap* (*PIC*). See Fig. 1(b) for a PUF-tree, which represents the same database of uncertain data as the UF-tree in Fig. 1(a).

Definition 1. The ***prefixed item cap (PIC)*** [13] of an item x_r in a transaction $t_j = \{x_1, \ldots, x_r, \ldots, x_h\}$ where $1 \leq r \leq h$—denoted as $\boldsymbol{I^{Cap}(x_r, t_j)}$—is

Table 1. A transactional database of uncertain data ($minsup = 1.1$)

TID	Transaction	Sorted transaction with infrequent items removed
t_1	$\{a{:}0.5,\ b{:}0.2,\ c{:}0.1,\ e{:}0.9,\ g{:}0.6\}$	$\{a{:}0.5,\ b{:}0.2,\ c{:}0.1,\ e{:}0.9,\ g{:}0.6\}$
t_2	$\{a{:}0.6,\ b{:}0.1,\ c{:}0.2,\ f{:}0.8,\ g{:}0.5\}$	$\{a{:}0.6,\ b{:}0.1,\ c{:}0.2,\ f{:}0.8,\ g{:}0.5\}$
t_3	$\{a{:}0.7,\ b{:}0.2,\ c{:}0.2,\ f{:}0.9\}$	$\{a{:}0.7,\ b{:}0.2,\ c{:}0.2,\ f{:}0.9\}$
t_4	$\{a{:}0.9,\ b{:}0.3,\ c{:}0.1,\ e{:}0.8,\ f{:}0.6\}$	$\{a{:}0.9,\ b{:}0.3,\ c{:}0.1,\ e{:}0.8,\ f{:}0.6\}$
t_5	$\{b{:}0.9,\ c{:}0.9,\ d{:}0.4\}$	$\{b{:}0.9,\ c{:}0.9\}$



(a) UF-tree **(b)** PUF-tree

Fig. 1. The UF-tree and PUF-tree for uncertain data in Table 1 when $minsup = 1.1$

defined as the product of (i) $P(x_r, t_j)$ and (ii) the highest existential probability value M_1 of items from x_1 to x_{r-1} in t_j (i.e., in the *proper prefix* of x_r in t_j):

$$I^{Cap}(x_r, t_j) = \begin{cases} P(x_r, t_j) \times M_1 & \text{if } h > 1 \\ P(x_1, t_j) & \text{if } h = 1 \end{cases} \tag{2}$$

where $M_1 = \max_{1 \leq q \leq r-1} P(x_q, t_j)$. □

As the PUF-growth algorithm mines frequent patterns by taking advantage of the tree structure to restrict the computation of upper bounds of expected support to the highest existential probability among items in the *prefix* of x via the use of the PIC, direct benefits include fewer false positives and shorter mining time because fewer projected databases need to be extracted and less work is required in a third scan of the uncertain data.

3 Our Proposed Tightened Prefixed Pattern Cap

Recall from Sect. 2 that each node in a PUF-tree keeps $I^{Cap}(x_r, t_j)$, which serves as an upper bound of expected support to any pattern formed from t_j with suffix x_r. See Example 1.

Example 1. Consider t_1 in Table 1. If $X = \{a, b, c, e\}$, then $I^{Cap}(e, t_1) = P(e, t_1) \times M_1 = 0.9 \times 0.5 = 0.45$.

This PIC also serves as an upper bound to the expected support of $\{a, e\}$, $\{b, e\}$, $\{c, e\}$, $\{a, b, e\}$, $\{a, c, e\}$, $\{b, c, e\}$ and $\{a, b, c, e\}$. While this upper bound is tight for short patterns like $\{a, e\}$ having $P(\{a, e\}, t_1) = 0.45$, it becomes loose for long patterns like $\{a, b, e\}$ having $P(\{a, b, e\}, t_1) = 0.09$ and $\{a, b, c, e\}$ having $P(\{a, b, c, e\}, t_1) = 0.009$. □

Observed from the above example, an upper bound based on PIC may not be too tight when dealing with long patterns mined from long transactions of uncertain data. In many real-life situations, it is not unusual to have long patterns to be mined from long transactions of uncertain data. To tighten the upper bound for patterns of all cardinality k (i.e., k-itemsets for $k \geq 2$), we propose a new concept of *tightened prefixed pattern cap* (*TPC*). The key idea is to keep track of a new value—a *"silver" value*—which is the second highest probability value M_2 in the prefix of t_j. Every time a frequent extension ($k > 2$) is added to the suffix item x_r, this "silver" value is used. As a preview, each node in the corresponding tree structure contains (i) an item x_r, (ii) its PIC, and (iii) its "silver" value. See the following definitions, examples, and observations.

Definition 2. Let (i) $t_j = \{x_1, \ldots, x_r, \ldots, x_h\}$ where $1 \leq r \leq h$, (ii) $X = \{y_1, y_2, \ldots, y_k\}$ is a k-itemset in t_j such that $y_k = x_r$, and (iii) M_2 denoting the "silver" value be the second highest existential probability value of items from x_1 to x_{r-1} in t_j (i.e., in the *proper prefix* of x_r in t_j). Then, the ***tightened prefixed pattern cap (TPC)*** is defined as follows:

$$TPC(X, t_j) = \begin{cases} I^{Cap}(x_r, t_j) & \text{if } k \leq 2 \\ I^{Cap}(x_r, t_j) \times \prod_{i=3}^{k} M_2 = I^{Cap}(x_r, t_j) \times M_2^{k-2} & \text{if } k \geq 3 \end{cases} \quad (3)$$

where $I^{Cap}(x_r, t_j)$ as defined in Definition 1. □

Example 2. Revisit Example 1 by reconsidering t_1 in Table 1. If $X = \{a, b, c, e\}$, then $TPC(X, t_1) = I^{Cap}(e, t_1) \times M_2^2 = 0.45 \times 0.2^2 = 0.018$, which is much closer to its $P(X, t_1) = 0.009$ when compared with the old bound of 0.45 provided by $I^{Cap}(e, t_1)$.

Similarly, if $X = \{a, b, e\}$, then $TPC(X, t_1) = I^{Cap}(e, t_1) \times M_2 = 0.45 \times 0.2 = 0.09$, which is as tight as its $P(\{a, b, e\}, t_1) = 0.09$. If $X = \{a, e\}$, then $TPC(X, t_1) = I^{Cap}(e, t_1) = 0.45$, which again is as tight as its $P(\{a, e\}, t_1) = 0.45$. □

Definition 3. The ***cap of expected support expSup^{Cap}(X)*** of a pattern $X = \{y_1, \ldots, y_k\}$ (where $k > 1$) is defined as the sum (over all n transactions in a database) of all the TPCs of y_k in all the transactions that contain X, i.e., $expSup^{Cap}(X) = \sum_{j=1}^{n} \{TPC(X, t_j) \mid X \subseteq t_j\}$. □

Observation 1. Based on Definition 3, $expSup^{Cap}(X)$ serves as an *upper bound* to the expected support of X, i.e., $expSup(X) \leq expSup^{Cap}(X)$. Hence, if $expSup^{Cap}(X) < minsup$, then X cannot be frequent. Conversely, if X is a frequent pattern, then $expSup^{Cap}(X)$ must be $\geq minsup$. Such a *safe/sound condition*—with respect to $expSup^{Cap}(X)$ and *minsup*—can be safely applied to mining all frequent patterns for data analytics. □

Observation 2. The expected support $expSup(X)$ satisfies the *downward closure property* [1] as $expSup(X) \leq expSup(Y)$ for all $Y \subset X$. So, (i) $expSup(X) \geq minsup$ implies $expSup(Y) \geq minsup$, and (ii) $expSup(Y) < minsup$ implies $expSup(X) < minsup$. □

Observation 3. The cap of expected support $expSup^{Cap}(X)$ of any pattern X based on the TPC does *not* always satisfy the downward closure property. As an example, $expSup^{Cap}(\{a, b, c\}) = 0.077 < 0.0828 = expSup^{Cap}(\{a, b, c, e\})$ in Table 1. □

Observation 4. For special cases where X and its subset Y sharing the same suffix item (e.g., $Y = \{a, b, e\} \subset \{a, b, c, e\} = X$ sharing the suffix item e), the *cap of expected support* based on the TPC satisfies the downward closure property. We call this property the *partial* downward closure property. □

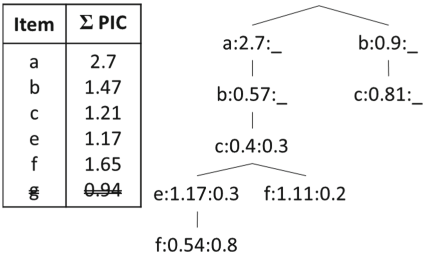
4 Our Proposed PUF*-tree Structure

Given that the TPC provides a tighter upper bound/cap to the expected support, we propose a new tree structure called *PUF*-tree* to efficiently capture contents of uncertain data so that the TPC can be computed based on Eq. (3) using the PIC and the "silver" value M_2. Specifically, each node in this PUF*-tree structure contains (i) an item x_r, (ii) its PIC, and (iii) its M_2. See Fig. 2.

The process of constructing a PUF*-tree can be described as follows. With the first scan of the database of uncertain data, we find all distinct frequent items and construct a header table called an *item-list* to store only frequent items in some consistent order (e.g., canonical order) to facilitate tree construction. Then, the PUF*-tree is constructed with the second database scan in a fashion similar to that of the FP-tree [5]. A key difference is that, when inserting a transaction item, we compute both its PIC and M_2 values. The item is then inserted into the PUF*-tree according to the ordering in the item-list. If a node containing that item already exists in the tree path, we update (i) its PIC by summing the computed PIC value with the existing one and (ii) its M_2 value by taking the *maximum* between the computed M_2 value and the existing one. Otherwise, we create a new node with the computed PIC and M_2 values. For a better understanding of the PUF*-tree construction, see Example 3.

Example 3. Consider the database of uncertain data in Table 1, and let the user-specified support threshold *minsup* be set to 1.1. Let the item-list follow the alphabetical ordering of items. After the first database scan, the contents of the item-list after computing the expected supports of all items and after removing infrequent items (e.g., item d) are $\langle a$:2.7, b:1.7, c:1.5, e:1.7, f:2.3, g:1.1$\rangle$.

With the second database scan, we insert only the frequent items of each transaction (with their respective PIC and M_2 values) in the ordering of the item-list. For instance, when inserting transaction $t_1 = \{a$:0.5, b:0.2, c:0.1, e:0.9, g:0.6$\}$, items a, b, c, e and g—with their respective PIC and (if appropriate) M_2 values such as $\langle$0.5, _$\rangle$ for a, $\langle 0.2\times0.5 = 0.1$, _$\rangle$ for b, $\langle 0.1\times0.5 = 0.05$, 0.2$\rangle$



Item	Σ PIC
a	2.7
b	1.47
c	1.21
e	1.17
f	1.65
~~g~~	~~0.94~~

Fig. 2. Our PUF*-tree for uncertain data in Table 1 when *minsup* = 1.1

for c, $\langle 0.9{\times}0.5 = 0.45,\ 0.2\rangle$ for e, $\langle 0.6{\times}0.9 = 0.54,\ 0.5\rangle$ for g) are inserted in the PUF*-tree. As t_2 shares a common prefix $\langle a,\ b,\ c\rangle$ with an existing path in the PUF*-tree created when t_1 was inserted, (i) the PIC values of those items in the common prefix (i.e., a, b and c) are added to their corresponding nodes (e.g., 0.5+0.6 = 1.1 for a, 0.1+0.06 = 0.16 for b, 0.05+0.12 = 0.17 for c), (ii) the M_2 values of those items are checked against the existing M_2 values for their corresponding nodes, with only the maximum saved for each node (e.g., $\max\{0.2, 0.1\} = 0.2$ for c), and (iii) the remainder of the transaction (i.e., a new branch for items f and g) is inserted as a child of the last node of the prefix (i.e., as a child of c). Figure 2 shows the PUF*-tree after inserting all the transactions and pruning those items with infrequent extensions (e.g., item g because its $expSup^{Cap}(\{g\})$)—provided by the total TPC value—is less than the user-specified *minsup*. See Observation 6. Similar to other tree structures for frequent pattern mining (e.g., FP-tree), our PUF*-tree maintains horizontal node traversal pointers, which are not explicitly shown in the figures for simplicity. □

With the aforementioned PUF*-tree construction process, we observe the following.

Observation 5. Although we arranged all items in canonical order when inserting them into the PUF*-tree in Example 3, we could also use other orderings (e.g., descending order of expected support or occurrence counts). If we were to store items in descending order of occurrence counts, then *the number of nodes in the resulting PUF*-tree would be the same as that of the FP-tree* [5]. □

Observation 6. Since the PIC value in each node is the same as in the PUF-tree, we can similarly remove any item having the sum of PIC values (in the item-list) less than *minsup* because it is guaranteed to have no frequent extensions. (The horizontal node traversal pointers allow us to visit such nodes in the PUF*-tree in an efficient manner.) Hence, we can remove item g from the PUF*-tree in Example 3 in the same way as in the PUF-tree because the sum of PIC values of g is less than *minsup*. This *tree-pruning technique* saves mining time as it skips all k-itemsets (for $k \geq 2$) with suffix g as they are all infrequent. □

Observation 7. Based on the aforementioned PUF*-tree construction process, the PIC value in a node x in a PUF*-tree maintains the sum of PIC values of an item x for all transactions that pass through or end at x. Because common prefixes are shared, the *PUF*-tree becomes more compact than the UFP-tree* [2] and avoids having siblings (nodes with the same parent node) containing the same item but having different existential probability values. □

Observation 8. As the PUF*-tree captures all frequent items in every transaction of uncertain data and stores their PIC & M_2 values, frequent pattern mining based on the TPC computed using PIC & M_2 values ensures that no frequent patterns will be missed (i.e., *no false negatives*). □

Observation 9. Recall that the expected support of $X = \{x_1, \ldots, x_k\}$ is computed by summing $P(X, t_j)$ of every t_j (ref. Eq. (1)), where $P(X, t_j)$ is the product of the existential probability value of x_k with those of other items in the proper prefix of X, i.e., $P(X, t_j) = P(x_k, t_j) \times \left(\prod_{i=1}^{k-1} P(x_i, t_j)\right)$. Also, recall that PUF-growth [13] utilizes a PIC computed based on the existential probability value of x_k and the *single highest* existential probability value M_1 in its prefix: $I^{Cap}(x_k, t_j) = P(x_k, t_j) \times M_1$. In contrast, the TPC for X—computed based on the existential probability value of x_k and the *two highest* existential probability values M_1 & M_2 in its prefix—provides a *tighter upper bound* because the TPC tightens the bound as potentially frequent patterns are generated during the mining process with increasing cardinality of X, whereas the PIC has no such compounding effect: $P(X, t_j) \leq TPC(x_k, t_j) \leq I^{Cap}(x_k, t_j)$ because $\left(P(x_k, t_j) \times \prod_{i=1}^{k-1} P(x_i, t_j)\right) \leq \left(P(x_k, t_j) \times M_1 \times M_2^{k-2}\right) \leq (P(x_k, t_j) \times M_1)$. □

5 Our Proposed PUF*-growth Algorithm

Here, we propose a scalable data analytics algorithm called *PUF*-growth*, which mines frequent patterns in a pattern-growth fashion from our PUF*-tree structure that captures uncertain data. Recall from Sect. 4 that the construction of a PUF*-tree is similar to that of a PUF-tree, except that "silver" values are additionally stored. Thus, the basic operation in PUF*-growth for mining frequent patterns is to construct a projected database for each potential frequent pattern and recursively mine its potentially frequent extensions.

Once an item x is found to be potentially frequent, the existential probability of x must contribute to the expected support computation for every pattern constructed from its $\{x\}$-projected database (denoted as DB_x). Hence, $expSup^{Cap}(\{x\})$ based on the TPC is guaranteed to be the upper bound of the expected support of any pattern with suffix x due to Observation 9. Theoretically, this implies that the complete set of patterns with suffix x can be mined based on the partial downward closure property (Observation 4). Practically, we can directly proceed to generate all potentially frequent patterns from the PUF*-tree due to the following observation.

Observation 10. Let (i) X be a k-itemset (where $k > 1$) with $expSup^{Cap}(X) \geq$ *minsup* in the database and (ii) Y be an itemset in the X-projected database (denoted as DB_X). Then, $expSup^{Cap}(Y \cup X)$ in the original database $\geq minsup$ if and only if $expSup^{Cap}(Y)$ in all the transactions in $DB_X \geq minsup$. □

Based on Observations 4 and 10, we apply the PUF*-growth algorithm to our PUF*-tree for generating only those k-itemsets (where $k > 1$) with caps of expected support $\geq$ *minsup*. Similar to UFP-growth [2] and PUF-growth [13], our PUF*-growth mining process may generate some false positives at the end of the second database scan, and all these false positives will be filtered out with the third database scan. Hence, our PUF*-growth is guaranteed to return *all* and *only those* frequent patterns with *neither* false positives *nor* false negatives.

Example 4. The PUF*-growth algorithm mines extensions of every item in the item-list/header. For example, with when $minsup = 1.1$, the $\{f\}$-conditional tree is constructed by extracting the tree paths $\langle a$:2.7:_, b:0.57:_, c:0.4:0.3, e:1.17:0.3, f:0.54:0.8$\rangle$ and $\langle a$:2.7:_, b:0.57:_, c:0.4:0.3, f:1.11:0.2$\rangle$. When projecting these two paths, PUF*-growth computes the cap of expected support for each item in the projected database using the PIC and M_2 values from all f nodes in the original tree.

This $\{f\}$-conditional tree is then used to generate (i) all 2-itemsets containing item f and (ii) their further extensions by recursively constructing projected databases from them. For all k-itemsets (where $k \geq 3$) that are generated, the cap of expected support is multiplied by the M_2 value. Consequently, for cardinality $k = 2$, potentially frequent patterns $\{a, f\}, \{b, f\}, \{c, f\}$ & $\{e, f\}$ are generated because all of them have their caps of expected support equal to $0.54 + 1.11 = 1.65$. However, unlike PUF-growth, no potentially frequent patterns of higher cardinality are generated with this suffix. For instance, we do not generate $\{a, b, f\}, \{a, c, f\}$ & $\{b, c, f\}$ because their caps of expected support equal to $(0.54 \times 0.8) + (1.11 \times 0.2) < minsup$. We also do not generate $\{a, e, f\}, \{b, e, f\}$ & $\{c, e, f\}$ because their caps are even lower.

Patterns ending with items e, b and c can then be mined in a similar fashion. The complete set of potentially frequent patterns generated by PUF*-growth includes $\{a, b\}$:1.29, $\{a, c\}$:1.21, $\{b, c\}$:1.21, $\{a, e\}$:1.17, $\{b, e\}$:1.17, $\{c, e\}$:1.17, $\{a, f\}$:1.65, $\{b, f\}$:1.65, $\{c, f\}$:1.65, & $\{e, f\}$:0.54. All of them are then checked against the database to find those truly frequent ones (after the third scan). □

As shown in Example 4, PUF*-growth finds a complete set of patterns from a PUF*-tree without any false negatives. In addition, with the small concession of storing one extra value in each node (i.e., the "silver" value), PUF*-growth does so while generating fewer false positives than PUF-growth. In much larger databases this effect has a huge impact on the number of the false positives generated and thus directly results in lower runtimes.

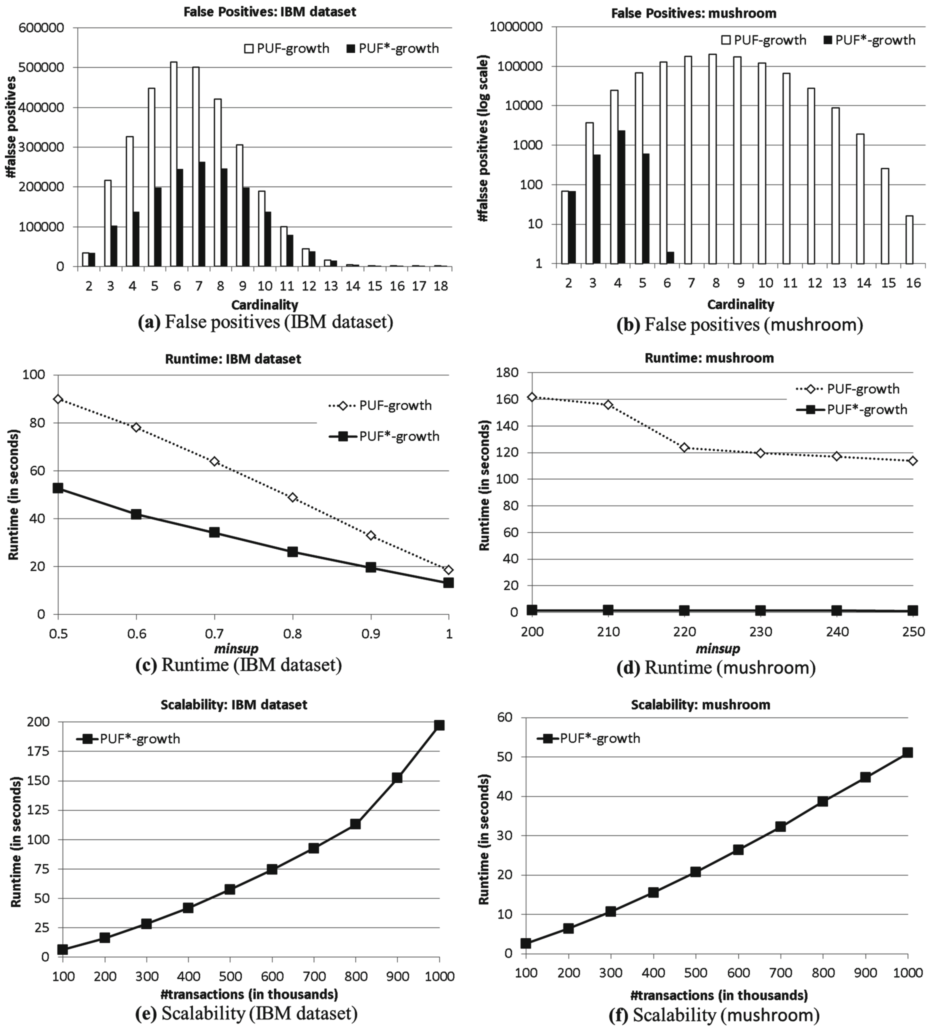
6 Evaluation Results

We compared the performances of our PUF*-growth algorithm with the existing PUF-growth [13] algorithm, which was shown [13] to outperform UF-growth [11], UFP-growth [2] and UH-Mine [2]. We used both synthetic and real-life datasets for our tests. The synthetic datasets, which are generally sparse, were generated within a domain of 1000 items by the data generator developed at IBM Almaden Research Center [1]. We also considered several real-life datasets such as mushroom, retail and kosarak. We assigned a (randomly generated) existential probability value from the range (0,1] to each item in every transaction in these datasets. The name of each dataset indicates some characteristics of the dataset. For example, the dataset u100K_5L_10_100 contains 100 K transactions with average transaction length of 5, and each item in a transaction is associated with an existential probability value that lies within a range of [10 %, 100 %]. Due to space constraints, we present here only some results on the above datasets.

All programs were written in C++ and ran in a Linux environment on an Intel Core i5-661 CPU with 3.33 GHz and 7.5 GB ram. Unless otherwise specified, runtime includes CPU and I/Os for item-list construction, PUF*-tree construction, mining, and false-positive removal. While the number of false positives generated at the end of the second database scan may vary, all algorithms (ours and others) produce the same set of truly frequent patters at the end of the mining process. The results shown in this section are based on the average of multiple runs for each case. In all experiments, *minsup* was expressed in terms of the absolute support value, and all trees were constructed using the ascending order of item value.

6.1 False Positives

The existing PUF-growth algorithm [13] and our PUF*-growth algorithm both generated some false positives. Their overall performances depend on the number of false positives generated. In this experiment, we measured the number of false positives generated by both algorithms for fixed values of *minsup* with different datasets. Due to space constraints, we present results using one *minsup* value for each of the two datasets (i.e., u100K_5L_10_100) and mushroom_50_60 in Fig. 3(a)–(b). In general, PUF*-growth was observed to remarkably reduce the number of false positives when compared with PUF-growth. The primary reason of this improvement is that the upper bounds for the PUF*-growth algorithm are much tighter than PUF-growth for patterns of higher cardinality k (where $k > 2$), and thus fewer potentially frequent patterns are generated and subsequently fewer false positives. As shown in Fig. 3(a), PUF*growth generated around 50 % of the false positives generated by PUF-growth. Moreover, when existential probability values were distributed over a narrow range with a higher *minsup* as shown in Fig. 3(b), PUF*-growth generated (i) only 1.6 % of the false positives generated by PUF-growth when $3 \leq k \leq 6$ and (ii) *no* false positives when $k \geq 7$. In total, PUF*-growth generated only 0.36 % of false positives generated by PUF-growth. Furthermore, PUF*-growth required shorter runtimes than PUF-growth in every single experiment we ran.



(a) False positives (IBM dataset)

(b) False positives (mushroom)

(c) Runtime (IBM dataset)

(d) Runtime (mushroom)

(e) Scalability (IBM dataset)

(f) Scalability (mushroom)

Fig. 3. Experimental results

6.2 Runtime

Recall that PUF-growth was shown [14] to outperform UH-Mine [13] and subsequently UFP-growth [2]. Hence, we compared our PUF*-growth algorithm with PUF-growth. Figure 3(c)–(d) show that PUF*-growth required shorter runtimes than PUF-growth for datasets mushroom_50_60 and u100K_5L_10_100. The primary reason is that, even though PUF-growth finds all frequent patterns when mining an extension of X, it may suffer from the high computation cost of generating unnecessarily large numbers of potentially frequent patterns as it only uses $P(x_r, t_j)$ and the single highest existential probability value M_1 in the prefix of x_r in t_j in its PIC calculation. This allows large numbers of potentially frequent patterns of high cardinality to be generated with similar expected support cap values to those of low cardinality having the same suffix item. The use of the self-product of M_2 in PUF*-growth ensures that those patterns with high cardinality are never generated due to their expected support caps being much closer to the expected support. This effect becomes more pronounced with lower *minsup* values, widening the gap in runtimes even further between the two algorithms. Since the TPC calculation in PUF*-growth becomes closer to the true expected support value as the cardinality of potentially frequent patterns under consideration is increased, lower *minsup* values have a much smaller effect on increasing run-times in PUF*-growth than in PUF-growth.

6.3 Scalability

With high volumes of high-variety, high-veracity and valuable data that can be collected and transmitted at high velocity, it becomes important to have a scalable algorithm to analyze these data. To test the scalability of PUF*-growth, we applied the algorithm to mine frequent patterns from datasets with increasing size. The experimental results presented in Fig. 3(e)–(f) demonstrate that our algorithm (i) is scalable with respect to the number of transactions and (ii) can mine high volumes of uncertain data within a reasonable amount of time.

The experimental results show that our PUF*-growth algorithm effectively mines frequent patterns from uncertain data irrespective of distribution of existential probability values (whether most of them have low or high values and whether they are distributed into a narrow or wide range of values).

7 Conclusions

In this paper, we proposed a scalable data analytics algorithm called *PUF*-growth* to mine frequent patterns from uncertain data. The algorithm first constructs the *PUF*-tree structure* to capture important information from uncertain data. It then finds all potentially frequent patterns—i.e., patterns with upper bounds (based on the *tighten prefixed pattern cap* (*TPC*)) to expected support $\geq$ the user-defined *minsup* threshold—from this PUF*-tree structure. These mined potentially frequent patterns contain *all* truly frequent patterns (i.e., *no* false negatives) as well as some false positives (i.e., patterns with upper bounds to expected support $\geq$ *minsup* but with expected support $<$ *minsup*). Fortunately, PUF*-growth reduces the number of false positives by obtaining tight upper bounds to expected supports. Finally, PUF*-growth then checks each potentially frequent pattern to eliminate the false positives. Experimental results show the effectiveness of our PUF*-growth algorithm in mining frequent patterns from uncertain data for scalable data analytics.

Acknowledgments. This project is partially supported by NSERC (Canada) and University of Manitoba.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: VLDB 1994, pp. 487–499 (1994)
2. Aggarwal, C.C., Li, Y., Wang, J., Wang, J.: Frequent pattern mining with uncertain data. In: ACM KDD 2009, pp. 29–37 (2009)
3. Calders, T., Garboni, C., Goethals, B.: Efficient pattern mining of uncertain data with sampling. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010, Part I. LNCS (LNAI), vol. 6118, pp. 480–487. Springer, Heidelberg (2010)
4. Chui, C.-K., Kao, B., Hung, E.: Mining frequent itemsets from uncertain data. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 47–58. Springer, Heidelberg (2007)
5. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM SIGMOD 2000, pp. 1–12 (2000)
6. Huan, J.: Frequent graph patterns. In: Liu, L., Tamer Özsu, M. (eds.) Encyclopedia of Database Systems, pp. 1170–1175. Springer, New York (2009)
7. Jiang, F., Leung, C.K.-S., MacKinnon, R.K.: BigSAM: mining interesting patterns from probabilistic databases of uncertain Big data. In: Peng, W.-C., Wang, H., Bailey, J., Tseng, V.S., Ho, T.B., Zhou, Z.-H., Chen, A.L.P. (eds.) PAKDD 2014 Workshops. LNCS (LNAI), vol. 8643, pp. 774–786. Springer, Heidelberg (2014)
8. Lakshmanan, L.V.S., Leung, C.K.-S., Ng, R.T.: Efficient dynamic mining of constrained frequent sets. ACM TODS **28**(4), 337–389 (2003)
9. Leung, C.K.-S.: Mining uncertain data. WIREs Data Mining Knowl. Discov. **1**(4), 316–329 (2011)
10. Leung, C.K.-S., Hao, B.: Mining of frequent itemsets from streams of uncertain data. In: IEEE ICDE 2009, pp. 1663–1670 (2009)
11. Leung, C.K.-S., Mateo, M.A.F., Brajczuk, D.A.: A tree-based approach for frequent pattern mining from uncertain data. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 653–661. Springer, Heidelberg (2008)
12. Leung, C.K.-S., Tanbeer, S.K.: Fast tree-based mining of frequent itemsets from uncertain data. In: Lee, S., Peng, Z., Zhou, X., Moon, Y.-S., Unland, R., Yoo, J. (eds.) DASFAA 2012, Part I. LNCS, vol. 7238, pp. 272–287. Springer, Heidelberg (2012)
13. Leung, C.K.-S., Tanbeer, S.K.: PUF-tree: a compact tree structure for frequent pattern mining of uncertain data. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013, Part I. LNCS (LNAI), vol. 7818, pp. 13–25. Springer, Heidelberg (2013)
14. Tong, Y., Chen, L., Cheng, Y., Yu, P.S.: Mining frequent itemsets over uncertain databases. PVLDB **5**(11), 1650–1661 (2012)
15. Wu, C., Yang, H., Zhu, J., Zhang, J., King, I., Lyu, M.R.: Sparse Poisson coding for high dimensional document clustering. In: IEEE BigData Conference 2013, pp. 512–517 (2013)
16. Yang, H., Lyu, M.R., King, I.: Efficient online learning for multitask feature selection. ACM TKDD **7**(2), art. 6 (2013)

Parallel Time Series Modeling - A Case Study of In-Database Big Data Analytics

Hai Qian[1](✉), Shengwen Yang[1], Rahul Iyer[1], Xixuan Feng[1], Mark Wellons[2], and Caleb Welton[1]

[1] Predictive Analytics Team, Pivotal Inc., Palo Alto, USA
hqian@pivotal.io
[2] Amazon.Com Inc., Seattle, USA

Abstract. MADlibis an open-source library for scalable in-database analytics. In this paper, we present our parallel design of time series analysis and implementation of ARIMA modeling in MADlib's framework. The algorithms for fitting time series models are intrinsically sequential since any calculation for a specific time t depends on the result from the previous time step $t-1$. Our solution parallelizes this computation by splitting the data into n chunks. Since the model fitting involves multiple iterations, we use the results from previous iteration as the initial values for each chunk in the current iteration. Thus the computation for each chunk of data is not dependenton on the results from the previous chunk. We further improve performance by redistributing the original data such that each chunk can be loaded into memory, minimizing communication overhead. Experiments show that our parallel implementation has good speed-up when compared to a sequential version of the algorithm and R's default implementation in the "stats" package.

Keywords: Parallel computation · Time series · Database management system · Machine learning · Big data · ARIMA

1 Introduction

Time series analysis plays an important part in econometrics, finance, weather forecasting, earthquake prediction and many other fields. For example, one of the most conspicuous data analytics task, stock price forecasting, falls into the category of time series analysis. Unlike other data analytics, time series data has a natural temporal ordering. And many time series modeling methods, such as autoregressive integrated moving average (ARIMA) [4] and Cox proportional hazards [7], therefore depend on sequential processing of time series data, which raises a challenge for the data-parallel implementation.

The first four authors contributed equally to this work.
Mark Wellons was an intern with Pivotal Inc. during the completion of this work.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 417–428, 2014.
DOI: 10.1007/978-3-319-13186-3_38

In this paper, we present our parallel implementation of ARIMA, a popular and important time series analysis model, in MADlib[14]. Although all algorithms for ARIMA modeling process the data sequentially, we can still split the data into multiple chunks, each containing consecutive parts of the time series. Each chunk needs the computation result from its neighboring chunk as initial values for its computation. Since the algorithm executes over multiple iterations, we can overcome this limitation by using the results of the preceding chunk from the previous iteration instead of that of the current iteration. The key idea is to take advantage of the iterative nature of the learning algorithms and rely on only local ordering, as illustrated in Fig. 1. At the time of convergence, there is no difference between the values from the previous iteration and the values from the current iteration.

Finally, it is important to note that the data in the database is not necessarily ordered. For fitting models like ARIMA, where the data has to be processed in the fixed order of time, one has to order the data in every iteration, which is time consuming. We avoid this by chunking, sorting, and re-distributing the data accross the segments so that a chunk of ordered data can be read into memory all at once in a segment. This technique not only avoids ordering the data repeatedly but also decreases both the I/O overhead and database function invocation overhead.

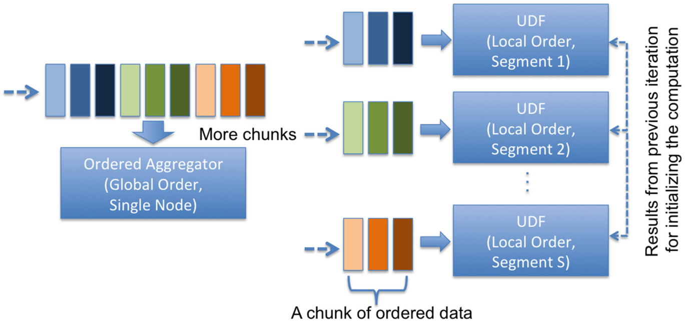


Fig. 1. Comparison between sequential and distributed designs. A segment is an independent database process in a shared-nothing distributed MPP (massively parallel processing) database.

The implementation is part of an open-source, scalable, in-database analytics initiative, MADlib [14], maintained by the Predictive Analytics Team at Pivotal Inc [1]. It provides data-parallel implementations of mathematical, statistical and machine-learning methods for structured and unstructured data.

1.1 Related Work

Parallel and Scalable Implementations. The problem of designing parallel algorithms has attracted much attention (see [3] for recent tutorials). Significant efforts have been spent on parallelizing various intricate machine learning algorithms, including k-means++ [2], support vector machines [16] and conditional random fields [5]. This work focuses on parallelizing time series analysis algorithms, where the training process requires a global ordering.

Machine Learning in Databases. Database management systems, being the best in data storage, operation, and analysis for many years, are also studied for data mining and machine learning on large datasets [9,12]. Ordonez [10] suggested sufficient statistics can be efficiently computed for an important set of models. Feng et al. [8] proposed an architecture linking an essential database programming model, user-defined aggregates, to convex optimization. These techniques, however, did not address time series analysis models in the context of databases.

1.2 What Is MADlib?

MADlibis an analytics platform developed by the Predictive Analytics Team at Pivotal Inc. (previously Greenplum). It can be deployed either onto the Greenplum database system (an industry-leading shared-nothing MPP database system) or the open-source PostgreSQL database system. The package itself is open-source and free. The MADlibproject was initiated in late 2010 from a research idea by Cohen et al. [6] who suggested a new trend of big data analytics requiring advanced (mathematical, statistical, machine learning), parallel and scalable in-database functions ("MAD" stands for "magnetic", "agile", and "deep" - see [6] for more details on each property).

By itself MADlib provides a pure SQL interface. To better facilitate data scientists from the R [11] community, there also exists a R front-end package called PivotalR [15]. On Greenplum database (GPDB) systems, MADlibutilizes the data parallel functionality. The calculation is performed in parallel on multiple segments of GPDB, and results from the segments are merged and then summarized on the master node. In many cases, multiple iterations of such calculations are needed. The core functionality for each iteration is implemented in C++ and Python is used to collate results from all iterations.

Although MADlibdoes not perform parallel computation on the open-source database system PostgreSQL, it is still valuable for processing large data sets that cannot be loaded into memory. For example, in this paper we will describe a comparison between our implementation of ARIMA and the 'arima' function in R's 'stats' package. For building an ARIMA model for a time series data set with the length 10^8, MADlibon PostgreSQL has about the same execution time as R, while consuming only 0.1 % of the memory used by R. Thus, MADlibprovides the ability of processing large data sets to open-source users for free.

MADlibhas various modules including linear, logistic, multinomial logistic regression, elastic-net regularization for linear and logistic regressions, k-means, association rules, cross validation, matrix factorization methods, LDA, SVD and PCA, ARIMA and many other statistical functions. A detailed user documentation is available online at http://doc.madlib.net. For this paper, we focus on our implementation for ARIMA in MADlib.

2 Implementation of ARIMA

In the next few subsections, we describe the algorithm used to solve the problem of maximization of partial log-likelihood to obtain the optimal values of the coefficients of ARIMA. Then, we point out the reason why it is difficult to make the algorithm run in-parallel. This is a common situation in all algorithms that fit ARIMA models. Next, we describe a generic solution to this problem that can be applied to time series problems. Then we describe a simple method to improve the performance of our algorithm. Finally, we discuss various ways to generalize our method to other algorithms.

2.1 The Algorithm

This section follows the design document for ARIMA on the MADlib website [13].

An ARIMA model is an auto-regressive integrated moving average model. An ARIMA model is typically expressed in the form

$$(1-\phi(B))Y_t = (1+\theta(B))Z_t, \tag{1}$$

where B is the backshift operator. The time t is from 1 to N.

ARIMA models involve the following variables:

1. The lag difference Y_t, where $Y_t = (1-B)^d(X_t - \mu)$.
2. The values of the time series X_t.
3. p, q, and d are the parameters of the ARIMA model. d is the differencing order, p is the order of the AR operator, and q is the order of the MA operator.
4. The AR operator $\phi(B)$.
5. The MA operator $\theta(B)$.
6. The mean value μ, which is set to zero when $d > 0$, or can be estimated by the ARIMA algorithm.
7. The error terms Z_t.

The auto regression operator models the prediction for the next observation as some linear combination of the previous observations. More formally, an AR operator of order p is defined as

$$\phi(B)Y_t = \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} \tag{2}$$

The moving average operator is similar, and it models the prediction for the next observation as a linear combination of the errors in the previous prediction errors. More formally, the MA operator of order q is defined as

$$\theta(B)Z_t = \theta_1 Z_{t-1} + \cdots + \theta_q Z_{t-q}. \tag{3}$$

We assume that

$$\Pr(Z_t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-Z_t^2/2\sigma^2}, \quad t > 0 \tag{4}$$

and that $Z_{-q+1} = Z_{-q+2} = \cdots = Z_0 = Z_1 = \cdots = Z_p = 0$.

The likelihood function L for N values of Z_t is then

$$L(\phi, \theta) = \prod_{t=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-Z_t^2/2\sigma^2} \tag{5}$$

so the log likelihood function l is

$$\begin{aligned} l(\phi, \theta) &= \sum_{t=1}^{N} \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-Z_t^2/2\sigma^2}\right) \\ &= \sum_{t=1}^{N} - \ln\left(\sqrt{2\pi\sigma^2}\right) - \frac{Z_t^2}{2\sigma^2} \\ &= -\frac{N}{2} \ln\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \sum_{t=1}^{N} Z_t^2. \end{aligned} \tag{6}$$

Thus, finding the maximum likelihood is equivalent to solving the optimization problem (known as the conditional least squares formation)

$$\min_{\theta, \phi} \sum_{t=1}^{N} Z_t^2. \tag{7}$$

The error term Z_t can be computed iteratively as follows:

$$Z_t = Y_t - F_t(\phi, \theta, \mu) \tag{8}$$

where

$$F_t(\phi, \theta, \mu) = \mu + \sum_{i=1}^{p} \phi_i (Y_{t-i} - \mu) + \sum_{i=1}^{q} \theta_i Z_{t-i} \tag{9}$$

Levenberg-Marquardt algorithm (LMA), also known as the damped least-squares (DLS) method, provides a numerical solution to the problem of minimizing a function, generally nonlinear, over the function's parameter space. These minimization problems arise especially in least squares curve fitting and nonlinear programming.

To understand LMA, it helps to know the gradient descent method and the Gauss-Newton method. On many "reasonable" functions, the gradient descent method takes large steps when the current solution is distant from the true solution, but is slow to converge when the current solution is close to the true solution. The Gauss-Newton method is much faster for converging when the current iterate is in the neighborhood of the true solution. The LMA tries to achieve the best of both worlds and combine the gradient descent step with the Gauss-Newton step in a weighted average. For iterates far from the true solution, the step favors the gradient descent step, but as the iterate approaches the true solution, the Gauss-Newton step dominates.

Like various numeric minimization algorithms, LMA is an iterative procedure. To start a minimization, the user has to provide an initial guess for the parameter vector, p, as well as some tuning parameters τ, ϵ_1, ϵ_2, ϵ_3, and $k_{\max}$. Let $Z(p)$ be the vector of calculated errors (Z_t's) for the parameter vector p, and let $J = (J_1, J_2, \ldots, J_N)^T$ be a Jacobian matrix.

A proposed implementation is shown in Algorithm 1.

```
**Input**: An initial guess for parameters $\boldsymbol{\phi}_0, \boldsymbol{\theta}_0, \mu_0$
**Output**: The parameters that maximize the likelihood $\boldsymbol{\phi}^*, \boldsymbol{\theta}^*, \mu^*$
$k \leftarrow 0$; $v \leftarrow 2$; $(\boldsymbol{\phi}, \boldsymbol{\theta}, \mu) \leftarrow (\boldsymbol{\phi}_0, \boldsymbol{\theta}_0, \mu_0)$;
Calculate $Z(\boldsymbol{\phi}, \boldsymbol{\theta}, \mu)$ with Eq. (9). // Vector of errors;
$A \leftarrow J^T J$ // The Gauss-Newton Hessian approximation;
$u \leftarrow \tau * \max_i(A_{ii})$ // Weight of the gradient-descent step;
$g \leftarrow J^T Z(\boldsymbol{\phi}, \boldsymbol{\theta}, \mu)$ // The gradient descent step;
stop $\leftarrow (\|g\|_\infty \leq \epsilon_1)$ // Termination Variable;
while not stop and $k < k_{\max}$ do
    $k \leftarrow k + 1$;
    repeat
        $\delta \leftarrow (A + u \times \mathrm{diag}(A))^{-1} g$ // Calculate step direction;
        if $\|\delta\| \leq \epsilon_2 \|(\boldsymbol{\phi}, \boldsymbol{\theta}, \mu)\|$ then // Change is too small to continue.
            stop $\leftarrow$ true;
        else
            $(\boldsymbol{\phi}_{new}, \boldsymbol{\theta}_{new}, \mu_{new}) \leftarrow (\boldsymbol{\phi}, \boldsymbol{\theta}, \mu) + \delta$ // A trial step;
            $\rho \leftarrow (\|Z(\boldsymbol{\phi}, \boldsymbol{\theta}, \mu)\|^2 - \|Z(\boldsymbol{\phi}_{new}, \boldsymbol{\theta}_{new}, \mu_{new})\|^2)/(\delta^T(u\delta + g))$ ;
            if $\rho > 0$ then // Trial step was good
                $(\boldsymbol{\phi}, \boldsymbol{\theta}, \mu) \leftarrow (\boldsymbol{\phi}_{new}, \boldsymbol{\theta}_{new}, \mu_{new})$ // Update variables;
                Calculate $Z(\boldsymbol{\phi}, \boldsymbol{\theta}, \mu)$ with Eq. (9); $A \leftarrow J^T J$; $g \leftarrow J^T Z(\boldsymbol{\phi}, \boldsymbol{\theta}, \mu)$;
                stop $\leftarrow (\|g\|_\infty \leq \epsilon_1)$ or $(\|Z(\boldsymbol{\phi}, \boldsymbol{\theta}, \mu)^2\| \leq \epsilon_3)$ ;
                $v \leftarrow 2$; $u \rightarrow u * \max(1/3, 1 - (2\rho - 1)^3)$;
            else// Trial step was bad
                $v \leftarrow 2v$; $u \leftarrow uv$;
            end
        end
    until stop or $\rho > 0$;
end
$(\boldsymbol{\phi}^*, \boldsymbol{\theta}^*, \mu^*) \leftarrow (\boldsymbol{\phi}, \boldsymbol{\theta}, \mu)$;
```

Algorithm 1. A proposed LMA implementation for fitting ARIMA model

Suggested values for the tuning parameters are $\epsilon_1 = \epsilon_2 = \epsilon_3 = 10^{-15}, \tau = 10^{-3}$ and $k_{\max} = 100$.

The Jacobian matrix $J = (J_1, J_2, \ldots, J_N)^T$ requires the partial derivatives, which are

$$J_t = (J_{t,\phi_1}, \ldots, J_{t,\phi_p}, J_{t,\theta_1}, \ldots, J_{t,\theta_q}, J_{t,\mu})^T \tag{10}$$

Here the last term is present only when we want to estimate the mean value of the time series too. The iteration relations for J are

$$J_{t,\phi_i} = -\frac{\partial Z_t}{\partial \phi_i} = Y_{t-i} - \mu + \sum_{j=1}^{q} \theta_j \frac{\partial Z_{t-j}}{\partial \phi_i} = Y_{t-i} - \mu - \sum_{j=1}^{q} \theta_j J_{t-j,\phi_i}, \tag{11}$$

$$J_{t,\theta_i} = -\frac{\partial Z_t}{\partial \theta_i} = Z_{t-i} + \sum_{j=1}^{q} \theta_j \frac{\partial Z_{t-j}}{\partial \theta_i} = Z_{t-i} - \sum_{j=1}^{q} \theta_j J_{t-j,\theta_i}, \tag{12}$$

$$J_{t,\mu} = -\frac{\partial Z_t}{\partial \mu} = 1 - \sum_{j=1}^{p} \phi_j - \sum_{j=1}^{q} \theta_j \frac{\partial Z_{t-j}}{\partial \mu} = 1 - \sum_{j=1}^{p} \phi_j - \sum_{j=1}^{q} \theta_j J_{t-j,\mu}. \tag{13}$$

Note that the mean value μ is considered separately in the above formulations. If we do not want to estimate the mean value, μ will be simply set to 0. Otherwise, μ will also be estimated together with $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$. The initial conditions for the above equations are

$$J_{t,\phi_i} = J_{t,\theta_j} = J_{t,\mu} = 0 \quad \text{for } t \leq p, \text{ and } i = 1, \ldots, p; j = 1, \ldots, q, \tag{14}$$

because we have fixed Z_t for $t \leq p$ to be a constant 0 in the initial condition. Note that J is zero not only for $t \leq 0$ but also for $t \leq p$.

2.2 Problems in Parallelization

It is easy to see that Eqs. (8, 11, 12, 13) are difficult to parallelize. Each step computation uses the result from the previous step. Therefore, we have to scan through the data sequentially to compute the quatities in these equations, and we have to do this in every iteration. If the data set is large then it is time-consuming to use this algorithm.

2.3 Our Solution

In order to utilize the data parallel capability of MPP (massively parallel processing) database, we propose to split the whole time series data into a set of sorted chunks numbered from 1 to N, each containing a sequence of consecutive time series data of size K. During the same time, we re-distribute the chunks of data onto all segments of the MPP database so that we could process them in parallel.

Since the model fitting involves multiple iterations of computations, for any subset i except for the first one, we use the results of the $(i-1)$-th subset from the previous iteration as the initial values. The first subset's initial values are known to be 0. Thus in each iteration, the computation for each subset of data does not need to wait for the results from the previous subset. In this way, the model fitting computation can be done in parallel.

Further, we find that aggregating each subset of consecutive time series data into an array (i.e. a data chunk) can greatly simplify the implementation (for example, from a stateful window function implementation to a plain UDF implementation) and accelerate the computation (mainly due to the reduction of I/O overhead and function call overhead). Furthermore, the value of size K is chosen in such a way that each chunk of data can be completely loaded into the available memory.

The following SQL script shows how we split and redistribute the data according to the above proposals.

```
-- redistribute the consecutive time series data into a same segment
create temp table dist_table as
select
  ((tid - 1) / chunk_size)::integer + 1 as distid, tid, tval
from input_table
distributed by (distid)

-- insert the preceding p data points for each chunk
insert into dist_table
select o.distid + 1, tid, NULL, tval
from (
  select distid, max(tid) maxid
  from dist_table
  where distid <> (select max(distid) from dist_table)
  group by distid
) q1, dist_table o
where q1.distid = o.distid and q1.maxid - o.tid < p

-- aggregate each chunk into an array to avoid repeated ordering
-- and communication overhead
create temp table work_table as
select
  distid, (distid - 1) * chunk_size + 1 as tid,
  array_agg(tval order by tid) as tvals
from dist_table
group by distid
distributed by (distid)
```

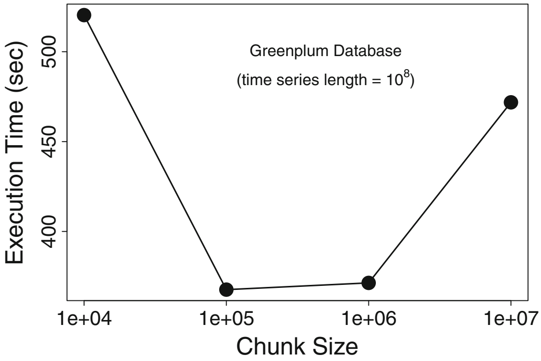


Fig. 2. We fit a time series with length of 10^8 using different chunk sizes. The execution times are around $400 \sim 500$ s, and are quite stable.

Our experiments show that the convergence of our implementation is very good. The ‘chunk size’ parameter does not have a significant impact on the performance as long as it is not too small, which can make the chunking meaningless, or not too large, which can make the data re-distribution difficult. As is shown in Fig. 2, the execution times are around 400 $\sim$ 500 s for different chunk sizes. The stable execution time for different chunk sizes makes it easier for the user to choose the proper parameters for the algorithms.

3 Experimentation

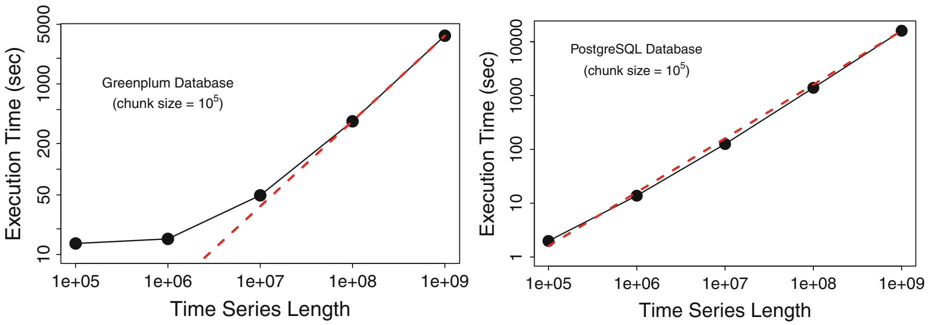
In this section, we present a set of experiments to measure the performance of our parallel implementation of ARIMA.

Configuration. We did our experiments on a DCA (Data Computation Appliance) produced by EMC Corporation, containing a Greenplum database system installed with 48 segments. The data sets that we used for the experiments are generated by R’s “arima.sim”. We generated multiple time series with different lengths. The data set with the length 10^9 is too large to be generated by R directly. Instead we first generated 10 pieces of time series with the lengths 10^8, and then assemble them together to form the complete time series.

3.1 Scalability

First, we measure the execution time of our implementation applied onto time series with different lengths. We run the tests in both Greenplum database and PostgreSQL database.

As is shown in Fig. 3, the execution time for large data sets is almost linear with respect to the length of the time series. For smaller data sets, the communication overhead between the multiple segments has a negative impact and the execution time is larger than the time for a pure linear execution time.



(a) (b)

Fig. 3. (left) We fit time series with MADlib’s ARIMA function on GPDB (left) and PostgreSQL(right). The chunk size in both cases is 10^5. The red dashed line is the fit to $t = \alpha l$, where t is the execution time, l is the length of the time series and α is a constant (Color figure online).

PostgreSQL database does not have the overhead of merging results from multiple segments, and thus the execution time, as shown in Fig. 3, is a linear function of the data size.

3.2 Total Runtime Comparisons

In Table 1, we compare the execution times of ARIMA model fitting in R and in MADlib on Greenplum database and PostgreSQL database. The execution time of MADlib's ARIMA on PostgreSQL is approximately the same as R (actually it is a little faster), but the memory usage is only a tiny fraction of R's "arima" function. This is because R loads all data into memory for processing, while the database systems esentially load one row of data into memory for processing and then proceed to the next row.

Although GPDB uses 48 segments, the speedup over PostgreSQL is about 3X to 4X. This is due to the communication overhead of communicating between multiple segments, especially the part where the data is loaded and re-distributed for sorting. If we compare the time taken for actual computation, GPDB is on average 17.6X faster than PostgreSQL.

Table 1. Here we compare the execution times of ARIMA model fitting in R and in MADlib on Greenplum Database and PostgreSQL database. The time series used to fit the ARIMA model has a length of 10^8. Note that running MADlib's ARIMA on Postgres is not only faster but also uses much less memory (0.1 %). Running this data set in R uses almost 70 % of the machine's memory (50 G memory). The iteration number for R is not available, because R's ARIMA does not output how many times it has iterated.

	MADlib on GPDB	MADlib on Postgres	R's arima function
Execution time (s)	364.4	1391.9	1964.4
Iteration number	29	29	N/A

3.3 Sensitivity of Number of Segments

For MADlib's ARIMA running on Greenplum database system, we also measured the execution time vs the number of segments used, which is shown in Fig. 4. Here, we use a time series with the length equal to 10^8. When the number of segments is less than 32, then the execution time decreases as more segments are added. However, increasing the number of segments beyond 32 will increase the execution time due to the increasing communication overhead between segments.

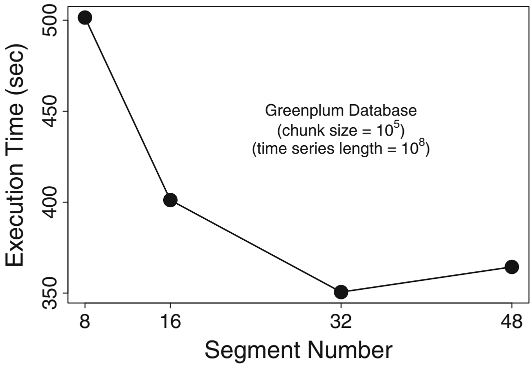


Fig. 4. We fit a time series with length of 10^8 using different numbers of segments in Greenplum database system.

4 Discussion and Conclusion

The notable methods that we used in the implementation of ARIMA in MADlibare: (1) split the data into chunks of consecutive data points and let each chunk use the result of the previous chunk from the previous iteration to initialize the calculation; (2) aggregate the chunk of data points into an array and redistribute the aggregated chunks accross segments so that each chunk of data can be loaded into memory and processed in one single function call by a segment. The first method makes it possible to parallelize the algorithm, and the second method greatly simplifies the implementation and improves the performance.

In this paper, we described our parallel implementation of ARIMA in MADlib and showed significant runtime improvements compared to serial implementations. It is easy to see that the above two methods can be easily generalized to other algorithms. The first method can be applied for any algorithm that requires a global ordering of the data. The second method can be used for improving the performance of any parallel algorithm. We aim to extend MADlibby generalizing this solution to parallelize other time series analysis algorithms.

References

1. Pivotal (2013). http://gopivotal.com/
2. Bahmani, B., Moseley, B., Vattani, A., Kumar, R., Vassilvitskii, S.: Scalable k-means++. Proc. VLDB Endowment **5**(7), 622–633 (2012)
3. Bekkerman, R., Bilenko, M., Langford, J.: Scaling up machine learning: parallel and distributed approaches. In: Proceedings of 17th ACM SIGKDD Tutorials, KDD '11 Tutorials, pp. 4:1–4:1 (2011)
4. Box, G.E., Pierce, D.A.: Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. J. Am. Stat. Assoc. **65**(332), 1509–1526 (1970)

5. Chen, F., Feng, X., Ré, C., Wang, M.: Optimizing statistical information extraction programs over evolving text. In: 2012 IEEE 28th International Conference on Data Engineering (ICDE), pp. 870–881. IEEE (2012)
6. Cohen, J., Dolan, B., Dunlap, M., Hellerstein, J.M., Welton, C.: MAD skills: new analysis practices for big data. Proc. VLDB Endowment **2**(2), 1481–1492 (2009)
7. Cox, D.: Regression models and life-tables. J. Roy. Stat. Soc. Ser. B (Methodological) **34**(2), 187–220 (1972)
8. Feng, X., Kumar, A., Recht, B., Ré, C.: Towards a unified architecture for in-rdbms analytics. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 325–336. ACM (2012)
9. Hellerstein, J.M., Ré, C., Schoppmann, F., Wang, D.Z., Fratkin, E., Gorajek, A., Ng, K.S., Welton, C., Feng, X., Li, K., Kumar, A.: The MADlib analytics library: or MAD skills, the SQL. Proc. VLDB **5**(12), 1700–1711 (2012)
10. Ordonez, C.: Building statistical models and scoring with udfs. In: Proceedings of the 2007 ACM SIGMOD International Conference on Management of data. ACM (2007)
11. R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna (2013)
12. Sarawagi, S., Thomas, S., Agrawal, R.: Integrating association rule mining with relational database systems: alternatives and implications, vol. 27. ACM (1998)
13. The Predictive Analytics Team at Pivotal Inc., Design document for MADlib (2013). http://madlib.net/design.pdf
14. The Predictive Analytics Team at Pivotal Inc. MADlib: an in-database analytics platform (2013). http://madlib.net
15. The Predictive Analytics Team at Pivotal Inc., PivotalR: an R front-end to both GPDB/Postgres and MADlib (2013). http://cran.r-project.org/web/packages/PivotalR/
16. Zhu, Z.A., Chen, W., Wang, G., Zhu, C., Chen, Z.: P-packsvm: parallel primal gradient descent kernel SVM. In: 2009 Ninth IEEE International Conference on Data Mining, ICDM'09, pp. 677–686. IEEE (2009)

A Biclustering-Based Classification Framework for Microarray Analysis

Baljeet Malhotra(✉), Daniel Dahlmeier, and Naveen Nandan

SAP Research and Innovation, CREATE Tower, 1 Create Way,
Singapore 138602, Singapore
{baljeet.malhotra,d.dahlmeier,naveen.nandan}@sap.com

Abstract. In recent years, microarrays have been shown to be an effective method for studying various biological processes, e.g., to improve our understanding of diseases such as cancer. In a typical situation, microarrays can be seen as large matrices in which rows and columns represent expression values of thousands of *genes* and tens of *conditions* such as samples from various patients. Several statistical techniques have been proposed in the literature to analyze the gene expression matrices. Towards that end, biclustering has been demonstrated to be one of the most effective methods for discovering gene expression patterns under various conditions. In this paper, we present a methodology to take advantage of the homogeneously expressed genes in biclusters to construct a classifier for sample class membership prediction. Our extensive experiments on 8 real cancer microarray datasets (4 diagnostic and 4 prognostic) show that our proposed classifier performed superior in both cancer diagnosis and prognosis, the latter of which was regarded quite difficult previously. Additionally, our results demonstrate that sample classification accuracy can serve as a good subjective quality measure for different types of biclusters, and hence as a tool to extrinsically evaluate the performance of various biclustering algorithms that produce those biclusters.

Keywords: Biclustering · Classification · Microarray analysis

1 Introduction

The advance of high-throughput hybridization microarray technology provides the opportunity to measure the expression levels of thousands of genes simultaneously, thus presenting a snapshot of the transcription levels within a cell. Such a technology enables researchers to look at cellular systems globally, for example, to improve our understanding on the disease related processes, yet also challenges us on effectively analyzing the vast volume of measured data such that key features of the cellular systems can be uncovered.

One of the major current applications of gene expression microarrays, particularly the high-density oligonucleotide arrays, such as the Affymetrix GeneChip oligonucleotide (Affy) arrays [1], is cancer diagnosis and prognosis. The underlying principle for this application (and many other applications) is that, two cells with dramatically different biological characteristics, such as a normal cell versus a cancerous cell from the same tissue, are expected to have different gene expression profiles. However, it is

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 429–440, 2014.
DOI: 10.1007/978-3-319-13186-3_39

important to realize that the majority of the active cellular mRNA is not affected by the differences. In other words, a dramatic biological difference does have a gene expression-level manifestation but the set of genes that is involved can be rather small. Microarray classification is to partition the arrays (also called conditions or samples) such that there are an extremely larger-than-expected number of genes sharply separating the classes.

Those genes that sharply separate the classes are referred to as informative or discriminatory genes, or *biomarkers*. Since these genes are expressed differentially under different conditions, such as samples from different individuals or organs or time points, they can be selected to compose gene expression profiles for the purpose of class prediction, upon the arrival of a new sample. Some early works on classification and class discovery exist [2] but their focus is on the sample partition (and not prediction). Other researchers have investigated two-way clustering of both genes and samples for defining sample classes and the class associated gene identification [1]. Note that sample partitioning requires homogeneous expression for all the genes while gene clustering assumes homogeneous expression of genes across all samples. With the increased understanding that not all genes express similarly in all samples, an alternate clustering framework, which produces local models, has been proposed to group genes and samples simultaneously, the so-called *biclustering*, which is also known (in several other areas of studies) as co-clustering, bi-dimensional clustering, and subspace clustering [3]. Towards that end, several biclustering algorithms have been proposed in the literature [4–6].

This paper, however, does not aim at proposing a new biclustering algorithm. Rather our study builds upon the state-of-the-art biclustering approaches to present a novel framework that effectively exploits biclusters for sample classification. The objective of the proposed classifier is to predict whether an unlabeled sample, possibly coming from a new patient or at a new time point, belongs to a particular class, e.g., cancerous genes. To the best of our knowledge, the proposed framework is the first to exploit biclusters for classification in microarray data. Our extensive experimental study using real microarray datasets reveal that good quality biclusters can be taken advantage for human cancer diagnosis and prognosis. Furthermore, these experimental results demonstrate that sample classification accuracy can serve as a good quality measurement for the discovered biclusters, disregarding their types.

The rest of paper is organized as follows: In the next section, we give some background on microarray data and some important concepts about biclusters. In Sect. 3, we present the details of our proposed framework that uses the genes in the biclusters for sample classification within the leave-one-out cross validation (LOOCV) scheme. Section 4 presents the cancer (diagnosis and prognosis) microarray datasets included in this study and our experimental results on them. Section 5 contains our discussion on both the classification framework and computational results. We conclude the paper in Sect. 6.

2 Microarray and Biclusters

In a typical situation, microarrays are seen as large matrices, commonly known as gene expression matrices, to represent thousands of genes along the rows and tens of conditions, e.g., samples from various patients or samples from various organs of a particular patient or samples taken at different time points, along the columns. Table 1 shows an example of a gene expression matrix in which the elements of the matrix, a_{ij}, represent the expression level of the *i*-th gene in the *j*-th condition. Each element, which is represented by a real number, is usually the logarithm of the relative abundance of the mRNA of the gene under the specific condition [3].

Table 1. An example of an n x m gene expression matrix. An element, a_{ij}, of the matrix represents the expression level of the i-th gene in the j-th sample.

	s_1	s_2	..	..	s_j	..	..	s_{m-1}	s_m
g_1	2.0	1.6	..	..	1.5	..	..	1.3	1.7
g_2	3.0	1.9	..	..	2.4	..	..	2.1	2.3
.	2.2	2.6	..	..	1.5	..	..	2.5	2.2
g_n	1.2	1.6	..	..	1.8	..	..	2.3	2.1

A number of clustering algorithms proposed in the literature can also be applied for the analysis of gene expression data. However, a large volume of gene expression data under multiple conditions often exhibits uncorrelated genes activity [3]. Therefore, clustering of genes (along the rows) using all conditions (along the columns) is often non-trivial and may produce undesired results. A similar problem exists for clustering of conditions using all genes.

Table 2. Three commonly known biclusters.

Constant			
3.2	1.6	3.6	3.9
3.2	1.6	3.6	3.9
3.2	1.6	3.6	3.9
3.2	1.6	3.6	3.9

Additive			
3.2	1.6	3.6	3.9
5.2	3.2	5.6	5.9
7.2	5.6	7.6	7.9
9.2	5.6	9.6	9.9

Multiplicative			
3.2	1.6	3.6	3.9
6.4	3.2	7.2	7.8
12.8	6.4	14.4	15.6
25.6	12.8	28.8	31.2

To solve the above problem, a number of algorithms have been proposed in the literature which allows simultaneous clustering of the rows and columns of a matrix, i.e., biclustering. The primary objective of these algorithms is to discover only those rows and columns, i.e., a subset of the given matrix or simply a bicluster, that exhibit some pattern. In the literature, there are several types of biclusters that have been defined and investigated [5–8]. Among them, constant [4], additive [6], and multiplicative [7] are three most commonly studied types. Examples for these three types are shown in Table 2. In a constant bicluster all the values (along the rows or columns) are constant. In our particular example, we show a constant-column bicluster. In an additive bicluster, the values (along the rows or columns) are incremented by adding a particular constant. In our particular example, we show an additive-column bicluster, in which the column values are incremented by adding 2. In a multiplicative bicluster, the matrix elements (along the rows or columns) are incremented with a multiplicative factor. In Table 2 we also show a multiplicative-column bicluster, in which the column values are incremented by a factor of 2.

Various algorithms for finding different types of biclusters have been proposed in the literature [3, 4, 6, 7] with few of them also conducting some theoretical studies on computational complexity. In these works, several bicluster quality measures have been examined, using methods such as value of the merit function defined for biclusters, statistical significance of the solution measured against the null hypothesis, and comparison against known solutions [3, 6]. Several methods emphasize the numerical quality of the identified biclusters, while others can incorporate existing biological knowledge such as gene functional annotation, gene co-regulation, and sample class membership [3]. For example, several works reviewed by Madeira and Oliveira [3] examine the relation between biclusters and sample class memberships [7, 8]. Unfortunately, it is unclear from the context on how biclusters can be used for sample class membership prediction, or sample classification, which is our main target in this study.

3 Classification Framework

The first step in our proposed framework is to generate biclusters that will be eventually used in our classification method. We will use A to denote the gene expression data matrix, which is an n x m matrix, with n being the number of genes and m being the number of samples. The entry a_{ij} records the expression level of the i-th gene in the j-th sample. Note that the order of genes and the order of samples are (normally) arbitrary and irrelevant in this study. Given a subset of genes $I \subseteq \{1, 2, \ldots, n\}$ and a subset of samples $J \subseteq \{1, 2, \ldots, m\}$, A_{IJ} denotes the sub-matrix of A by removing genes not in I and samples not in J.

Different from clustering (on genes or samples) which seeks for homogeneous gene expression (across all samples or genes, respectively), biclustering performs clustering in the two dimensions simultaneously, and thus to produce local models in contrast to global models produced by clustering. A bicluster is defined by a pair of a gene subset I and a sample subset J, expecting that genes in I have similar behavior across the samples in J. The notion of "similar behavior" can be characterized in several ways [3].

In this paper, we are particularly interested in two types of biclusters: constant and additive. A perfect constant bicluster is a sub-matrix A_{IJ} in which all entries are equal, that is, $a_{ij} = \mu$, for all $i \in I$ and $j \in J$. A perfect additive bicluster is a sub-matrix A_{IJ} with coherent values, which can be expressed as $a_{ij} = \mu + \alpha_i + \beta_j$, where α_i is the adjustment for the i-th gene and βj is the adjustment for the j-th sample. Clearly, a perfect constant bicluster is a special case of a perfect additive bicluster. Although these "ideal" biclusters can be found in some expression matrices, in real data, they are masked by noise.

3.1 Biclustering Methods

In this paper, we employ two algorithms, which are detailed next, for discovering two types of bi-clusters discussed above. In their seminal work [4], Cheng and Church defined a bicluster to have a high mean squared residue score, which is used as a measure of the coherence of the genes and samples in the bicluster. Given a bicluster A_{IJ}, set $a_{iJ} = \frac{1}{|J|}\Sigma_{j\in J}a_{ij}, a_{Ij} = \frac{1}{|I|}\Sigma_{i\in I}a_{ij}$, and $a_{IJ} = \frac{1}{|I||J|}\Sigma_{i\in I,j\in J}a_{ij}$. The residue score of entry a_{ij} in the bicluster is $a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}$. The mean squared residue score of A_{IJ}, denoted as H(I,J), is calculated as

$$H(I,J) = \Sigma_{i\in I,j\in J}\left(a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}\right)^2 \tag{1}$$

A_{IJ} is a δ-bicluster if $H(I,J) \leq \delta$ for some $\delta \geq 0$.

Clearly, a 0-bicluster is a perfect constant bicluster. Cheng and Church proposed several heuristic algorithms to discover δ-biclusters by removing rows and columns from the original matrix [4]. It is worth mentioning that their proposed algorithms have a tendency to find constant biclusters but not necessarily other types of biclusters such as additive or multiplicative. The particular algorithm we employ in this study is the Multiple Node Deletion (MND) algorithm, which iteratively removes genes whose contributing residue scores (defined as $\frac{1}{|J|}\Sigma_{j\in J}\left(a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}\right)^2$) are greater than $\alpha H(I,J)$, or when no such genes, only the gene with the highest such score, and samples whose contributing residue scores (defined as $\frac{1}{|I|}\Sigma_{i\in I}\left(a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}\right)^2$) are greater than $\alpha H(I,J)$, or when no such samples, only the sample with the highest such score, until $H(I,J)$ does not exceed δ. We denote this algorithm as MND(δ, α) for the particular pair of δ and α.

Recently, Liu and Wang proposed several algorithms for finding multiple (may be overlapping) maximum similarity biclusters [6], which include constant and additive ones. Given I and J, and a reference gene $i^* \in I$, finding a maximum similarity bicluster within I and J is to find a subset of genes $I' \subseteq I$ and a subset of samples $J' \subseteq J$ such that the distances between the reference gene i^* and genes in I' are minimized. In more details, define $d_{ij} = |a_{ij} - a_{i^*j}|$, and we want to discard those large d_{ij} 's to achieve the target bicluster. To this purpose, define the average difference as $\bar{d} = \frac{1}{|I||J|}\Sigma_{i\in I,j\in J}d_{ij}$. With a threshold α, a similarity matrix S_{IJ} is defined for A_{IJ} in which

$s_{ij} = 0$ if $d_{ij} \geq \alpha\bar{d}$, or otherwise $s_{ij} = 1 - d_{ij}/\alpha\bar{d} + \beta$ (where $\beta \geq 0$ is a bonus for small d_{ij}). Define the similarity score of the i-th gene in S_{IJ} as $s_{iJ} = \Sigma_{j\in J}\, s_{ij}$, the similarity score of the j-th sample in S_{IJ} as $s_{Ij} = \Sigma_{i\in I}\, s_{ij}$, and the similarity score of matrix A_{IJ} as $s_{IJ} = min\{min_{i\in I}\, s_{iJ}, min_{j\in J}\, s_{Ij}\}$. The particular algorithm we employ in this study is the MSB algorithm, which starts with the whole matrix A, repeatedly deletes the gene or the sample whose similarity score is the currently smallest to obtain $n + m - 1$ biclusters, and returns the one having the maximum similarity score s_{IJ} while its average similarity score $\bar{s}_{IJ} = \frac{1}{|I||J|}\Sigma_{i\in I,j\in J}s_{ij}$ is at least γ. We denote this algorithm as MSB(γ, α, β) for the three associated parameters.

3.2 Bicluster Generation

We use two algorithms, MND(δ, α) and MSB(γ, α, β), to generate biclusters in the (training) dataset, in which every sample has a known class membership. (The proposed framework is not dependent on any particular algorithm or the type of biclusters produced by it. However, the choice of an algorithm and the produced biclusters may influence the classification results as discussed later in Sect. 5.) For each algorithm, a range is pre-determined for every parameter, resulting in a number of distinct settings. First of all, we partition the expression matrix A (the training dataset) into sub-matrices which have the same set of genes but each contains only the samples from a particular class. (The number of such sub-matrices is equal to the number of classes in the dataset.) Separately or together, every setting of the biclustering algorithms is run on these sub-matrices to generate one bicluster per class.

In MND(δ, α) algorithm, δ was chosen based on a trial and error policy. With this policy, first, we ran MND(δ, α) algorithm (with some random δ value) on the whole dataset while noting down the residue score (H(I,J)) and the size of the generated bicluster. When the size of the gene set in the generated bicluster was less than half of the total number of genes, we chose that particular δ value as the initial value. Afterwards, we ran the algorithm multiple times while incrementally decreasing the δ value. For example, if the initial δ value for a given dataset was 700, we subsequently ran MND(δ, α) algorithm with δ values being 600, 500, and 400, and so on. Under this policy of parameter setting, the size of the gene set in the generated biclusters varied from 30 % to 10 % of the total number of genes in the whole dataset. Parameter α was kept at 1.1 throughout the experiments.

For MSB(γ, α, β) algorithm, parameters α, β and γ were chosen based on the original recommendations [6], where the authors suggested that $\alpha \in [0.2, 0.4]$, $\beta \in [0.0, 0.5]$ and $\gamma \in [\beta + 0.7, \beta + 0.9]$. We tried $\alpha = 0.3, 0.4$ and fixed β at 0.4 and γ at 1.2. With these settings the sizes of the gene sets of the generated biclusters varied from 20 % to 8 % of the total number of genes in the whole dataset. For each setting of (γ, α, β), (a maximum of) 5 reference genes were randomly selected from the gene pool.

3.3 Distance Calculation

The generated biclusters are considered as important and the genes in them are believed to strongly correlate to the sample classes. We take all the genes included in these top quality biclusters for calculating distances between a testing sample and the sample classes in the training dataset. Assume these genes form a set *I* and the training sample set is *J*. For each sample $j \in J$, the distance between testing sample s and sample *j* is calculated as the normalized *L1* distance using gene set *I*:

$$d_{L1}(s,j) = \frac{1}{|I|} \Sigma_{i \in I} \left| a_{is} - a_{ij} \right| \tag{2}$$

Note that a sample not in *J* has no distance to testing sample *s*. The distance between testing sample *s* and *a* sample class is defined as the average distance over all the samples in the class which have distances to *s*. The above *L1* distance can be substituted by other distance measures such as the Euclidean distance.

3.4 Classification and LOOCV Accuracy

Given all the discovered biclusters which define the gene set used in the distance calculation, whenever a testing sample arrives, we can calculate its distance to every sample class in the (training) dataset. The label of the closest class to the testing sample is taken as the predicted class label for the testing sample. In our experiment, we adopt the leave-one-out cross validation (LOOCV) scheme to calculate the classification accuracy. With each iteration, one sample is selected as the testing sample whose class membership is blinded to the classifier. Using the rest of the samples, biclustering algorithms are run to generate the target biclusters and the subsequent genes used in the distance calculation. One correct prediction is arrived when the predicted class label is the same as the true one. The LOOCV scheme iterates through all samples and the percentage of correct predictions is the LOOCV classification accuracy.

4 Experimental Results

4.1 Overview

All experiments were conducted in Matlab environment. We have implemented both algorithms, MND(δ, α) and MSB(γ, α, β), ourselves and thoroughly tested their correctness. For example, using the same datasets in their original paper, our implemented algorithms were tested on to generate biclusters, which were compared to the biclusters generated by the original authors. A test case is considered successful only if these two sets of biclusters matched with each other. The correctness is guaranteed by 100 % matching results in several test cases.

Afterwards, complete LOOCV sample classification was performed, using these two algorithms on several real cancer gene expression microarray datasets, for either diagnosis or prognosis purpose. In addition to the results of the two individual algorithms, we include results for a combination of the two algorithms which we refer to as MND + MSB. In this schema, biclusters are first generated using MND and MSB algorithms independently. The average distance is then computed as defined in Sects. 3.3 and 3.4 based on those biclusters (as discovered by MND and MSB algorithms) to determine the class membership.

The classification accuracies were reported and compared to the previously achieved best accuracies on the individual datasets.

4.2 Cancer Gene Expression Datasets

We have used 4 cancer diagnosis datasets and 4 prognosis datasets in our experiments, listed as follows.

Diagnostic Datasets. AML-ALL Leukemia dataset [9] consists of 72 samples in two classes: acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). The gene expression levels were measured using Affymetrix high-density oligonucleotide arrays containing 7,129 probes (from 6,817 human genes), 47 samples of ALL (38 B-cell ALL and 9 T-cell ALL) and 25 samples of AML. After filtering the dataset contains 3,571 genes. An LOOCV classification accuracy of 98.60 %, which is previously the best known, has been achieved on this dataset [10].

Lung Cancer dataset [11] is used for classification between malignant pleural mesothelioma (MPM) and adenocarcinoma (ADCA) of the lung. There are 181 tissue samples (31 MPM and 150 ADCA), on 12,533 genes. We do not have any known LOOCV result on this dataset.

The Brain tumor dataset consists of 50 high-grade glioma samples of which 28 are glioblastomas and 22 are anaplastic oligodendrogliomas [12]. Glioblastomas and anaplastic oligodendrogliomas samples are further classified into classic and non-classic tumors (14 and 14, 7 and 15, respectively). This dataset contains 12,625 genes. The best ever achieved LOOCV classification accuracy on this dataset is 80 % [13].

The Carcinomas dataset (U95a GeneChip) contains 174 samples in 11 classes: *prostate, bladder/ureter, breast, colorectal, gastroesophagus, kidney, liver, ovary, pancreas, lung adenocarcinomas,* and *lung squamous cell carcinoma*, which have 26, 8, 26, 23, 12, 11, 7, 27, 6, 14, and 14 samples, respectively [14]. Each sample originally contained 12,533 genes. We preprocessed the dataset as described in Su et al. [14] to include only those probe sets whose maximum hybridization intensity is ≥ 200 in at least one sample; subsequently, all hybridization intensity values ≤ 20 were raised to 20, and the values were log transformed. After preprocessing, we obtained a dataset of 9,183 genes. The best ever achieved LOOCV classification accuracy on this dataset is 93.6 % [13].

Prognostic Datasets. Breast Cancer (training) dataset [15] contains 78 patient samples, 34 of which are from patients who had developed distance metastases within 5 years (labeled as relapse), the rest 44 samples are from patients who remained healthy from the disease after their initial diagnosis for interval of at least 5 years (labeled as non-relapse). The original dataset contains 24,481 genes. Our version of dataset contains only 23,625 genes and 32 relapse samples and 44 non-relapse samples. The authors applied a selection scheme on genes and constructed a classifier based on the correlation coefficient to good prognosis templates and poor prognosis templates. The achieved LOOCV classification accuracy on this dataset is 73 %.

AML-Leukemia is a subset of the above described AML-ALL Leukemia dataset [9], which contains 7,129 probes (from 6,817 human genes) and 15 samples of AML. 8 treatments failed and the other 7 were successful. There is no LOOCV result for this dataset.

Central Nervous System dataset [16] is used to analyze the outcome of the treatment. Survivors are patients who are alive after treatment whiles the failures are those who, unfortunately, succumbed to their disease. The dataset contains 60 patient samples, 21 are survivors and 39 are failures, on 7,129 genes. The authors selected a subset of genes to construct a KNN-classifier and achieved a LOOCV accuracy of 78 %.

Prostate Cancer dataset [17] for prediction of clinical outcome contains 21 patients were evaluable with respect to recurrence following surgery with 8 patients having relapsed and 13 patients having remained relapse free (non-relapse) for at least 4 years. The dataset contains 12,600 genes. The authors selected a subset of genes to construct a KNN-classifier and achieved a LOOCV accuracy of 90 %.

4.3 LOOCV Classification Accuracies

Table 3 summarizes our results. The size column records the size of an individual dataset using the number of genes and the numbers of samples in all the classes. On each dataset, the previously best classification accuracy, to the best of our knowledge, is added for comparison purpose. We have three LOOCV classification accuracies, by only MND(δ, α) algorithm, by only MSB(γ, α, β), and by both of them jointly.

Table 3. The LOOCV classification accuracies achieved by our bicluster-based methods compared with the previously achieved best accuracies, on the eight cancer gene expression microarray datasets. The bold ones are the currently best LOOCV classification accuracies.

Dataset		Prev. Best	Our Accuracies (%)		
Name	Size	Accuracy (%)	MND	MSB	MND + MSB
ALL-AML Leukemia	3,371 x {47, 25}	(Cai et al., 2006) 98.60	97.22	**98.66**	97.22
Lung Cancer	12,533 x {150, 31}	–	**88.95**	84.53	88.95
Brain Tumor	12,625 x {14, 14, 7, 15}	(Cai et al., 2007) 80.00	88.00	**92.00**	88.00
Carcinomas	9,183 x {26, 8, 26,. ..}	(Cai et al., 2007) 93.60	91.95	**96.55**	91.95
Breast Cancer	23,625 x {32, 44}	(van't Veer et al., 2002) **73.00**	53.94	71.05	53.94
Leukemia-AML	7,129 x {8, 7}	–	**100.0**	100.0	100.0
Central Nervous System	7,129 x {39, 21}	(Pomeroy et al., 2002) **78.00**	40.00	53.33	40.00
Prostate Cancer	12,600 x {8, 13}	(Singh et al., 2002) 90.00	80.95	**95.23**	80.95

On six of the eight datasets for which we know the previously best LOOCV results, 4 of them are updated by our proposed method (in bold in Table 3). In particular, on the Carcinomas dataset, the detailed prediction results by MSB(0.3, 0.45, 1.2), using three randomly chosen reference genes, on all the classes are recorded in Table 4, where the correct predictions are in bold.

On the Breast Cancer dataset, our method performed competitively, 71.05 % versus 73.0 %. On the last Central Nervous System dataset, our method did not perform satisfactorily. It is worth noting that the small Leukemia-AML prognostic dataset was considered challenging for computational prognosis previously [9]. Our method achieved the perfect result on this small dataset. Overall, these results show the effectiveness of our proposed classification method.

Table 4. The detailed prediction results by MSB(0.3, 0.45, 1.2), using three randomly chosen reference genes, on all the classes in the Carcinomas dataset, where the correct predictions are in bold.

# Samples		P	BU	B	C	G	K	LI	O	PA	LA	LS
Prostate (P)	26	**26**										
Bladder/Ureter (BU)	8		**8**									
Breast (B)	26	1	1	**23**	1							
Colorectal (C)	23				**23**							
Gastroesophagus (G)	12					**11**					1	
Kidney (K)	11						**11**					
Liver (LI)	7							**7**				
Ovary (O)	27			1					**26**			
Pancreas (PA)	6									**6**		
Lung Adeno. (LA)	14										**14**	
Lung Squamous (LS)	14				1							**13**

5 Discussion

In the past several years, many sample classification algorithms have been proposed, most of which deal with the dimensionality issue (that is, tens of thousands of genes versus only tens of samples) through a step called gene selection. Essentially, various mechanisms have been set up to identify the most discriminatory genes, which express substantially different under different conditions, followed by classifier construction based on the selected genes.

Our classification method based on discovered biclusters may also be regarded as one of the kind, in that our selected genes are those that are included in the discovered biclusters. Nevertheless, our “gene selection” is very different from the existing ones in principle. Within the biclustering context, we partition the whole expression matrix into submatrices such that each contains only those samples in one sample class. The employed biclustering algorithms uncovered those genes that strongly correlate to the class. Therefore, using them in distance calculation is adequate and when the testing sample does belong to the particular class, the distance is expected to be small, or large otherwise.

5.1 Using Class-Dependent Genes Only

We have also tested the distance calculation between the testing sample and a particular class by using only those genes that are included in the biclusters generated for that class. The intention was similar in that when the testing sample belongs to this class, the calculated distance is expected small, or large otherwise. However, the computational results show that such a scheme is inferior, though not much, to the scheme of using all the occurring genes in the distance calculation. We thus chose not to report this set of results.

5.2 The Number of Biclusters

For each class, MND(δ, α) algorithm generated only one bicluster, and MSB(γ, α, β) algorithm generated no more than 5 biclusters. The percentage of genes occurring in these biclusters is roughly 30 % to 8 % of the total number of genes in the whole dataset. We have also tested to generate many more biclusters by changing the parameter setting, and then to select a few of them for distance calculation. It turned out that the latter did not perform better, while increasing the complexity.

5.3 The Size of Dataset

Most of the running time was consumed by the biclustering algorithms. The problem became more severe with increasing dataset size. With the tens of thousands of genes, the bottleneck is the class size, i.e., the number of samples in the particular class. We experienced some delays on several datasets, such as the diagnostic Lung Cancer dataset and the prognostic Breast Cancer dataset, of which the class sizes are relative large. Note that in the LOOCV scheme, the biclustering algorithms were run for a huge number of times. For example, on the diagnostic Lung Cancer dataset, each algorithm was run for 150 times on a dataset of size 12,533 × 149 and for 31 times on a dataset of size 12,533 × 30. When the class sizes are all relatively small, such as the diagnostic Carcinomas dataset (9,183 genes, the maximum class size is 27), the computation was quickly done.

6 Conclusion

In this paper, we presented formally a sample classification framework using the discovered biclusters. The extensive experiments demonstrated that the top ranked constant biclusters generated by two previously proposed algorithms can be taken advantage for the sample classification purpose. As a byproduct, the results demonstrated that sample classification accuracy can serve as an effective and biologically meaningful measurement for the bicluster quality, contrast to previously proposed measures that largely look at the numerical aspects matching to the bicluster definitions. Our proposed sample classification method is a generic framework, in that any biclustering algorithms for finding various types of biclusters can be plugged in.

Some of our future work subjects include investigating which type(s) of biclusters are more helpful for cancer diagnosis and prognosis purposes, better criteria for bicluster selection, better use of the genes included in the selected biclusters, a substantial comparative study to other most advanced classification algorithms, and the limit of our framework in terms of the dataset class number.

Acknowledgements. The authors would like to thank Guohui Lin for his guidance on this work and Zhipeng Cai for providing the author with a number of microarray datasets used in this study. This research was supported by the Economic Development Board and the National Research Foundation of Singapore.

References

1. Alon, U., Barkai, N., Notterman, D.A., et al.: Broad patterns of gene expressionrevealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Nat. Acad. Sci. USA **96**, 6745–6750 (1999)
2. Ben-Dor, A., Friedman, N., Yakhini, Z.: Class discovery in gene expression data. In: Proceedings of RECOMB 2001, pp. 31–38 (2001)
3. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Trans. Comput. Biol. Bioinf. **1**, 24–45 (2004)
4. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB 2000), pp. 93–103 (2000)
5. Cho, H., Dhillon, I.S., Guan, Y., Sra, S.: Minimum sum-squared residue cococlustering of gene expression data. In: Proceedings of the Fourth SIAM International Conference on Data Mining (2004)
6. Liu, X., Wang, L.: Computing the maximum similarity bi-clusters of gene expression data. Bioinformatics **23**, 50–56 (2007)
7. Klugar, Y., Basri, R., Chang, J.T., Gerstein, M.: Spectral biclustering of microarray data: Coclustering genes and conditions. Genome Res. **13**, 703–716 (2003)
8. Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. Bioinformatics **18**, 136–144 (2002)
9. Golub, T.R., Slonim, D.K., Tamayo, P., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science **286**, 531–537 (1999)
10. Cai, Z., Xu, L., Shi, Y., Salavatipour, M.R., Goebel, R., Lin, G.-H.: Using gene clustering to identify discriminatory genes with higher classification accuracy. In: Proceedings of IEEE The 6th Symposium on Bioinformatics and Bioengineering (IEEE BIBE 2006), Washington D.C., USA, pp. 235–242 (2006)
11. Gordon, G.J., Jensen, R.V., Hsiao, L.-L., et al.: Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. Cancer Res. **62**, 4963–4967 (2002)
12. Nutt, C.L., Mani, D.R., Betensky, R.A., et al.: Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. Cancer Res. **63**, 1602–1607 (2003)
13. Cai, Z., Goebel, R., Salavatipour, M.R., Lin, G.-H.: Selecting dissimilar genes for multi-class classification, an application in cancer subtyping. BMC Bioinform. **8**, 206 (2007)
14. Su, A.I., Welsh, J.B., Sapinoso, L.M., et al.: Molecular classification of human carcinomas by use of gene expression signatures. Cancer Res. **61**, 7388–7393 (2001)
15. van't Veer, L.J., Dai, H., van de Vijver, M.J., et al.: Gene expression profiling predicts clinical outcome of breast cancer. Nature **415**, 530–536 (2002)
16. Pomeroy, S.L., Tamayo, P., Gaasenbeek, M., et al.: Prediction of central nervous system embryonal tumor outcome based on gene expression. Nature **415**, 436–442 (2002)
17. Singh, D., Febbo, P.G., Ross, K., et al.: Gene expression correlates of clinical prostate cancer behavior. Cancer Cell **1**, 203–209 (2002)
18. Yang, J., Wang, W., Wang, H., Yu, P.: Enhanced biclustering on expression data. In: Proceedings of the Third IEEE Conference on Bioinformatics and Bioenginerring, pp. 321–327 (2003)

Parallel Implementation of a Density-Based Stream Clustering Algorithm Over a GPU Scheduling System

Marwan Hassani[1](✉), Ayman Tarakji[2], Lyubomir Georgiev[1,2], and Thomas Seidl[1]

[1] Data Management and Data Exploration Group, RWTH Aachen University, Aachen, Germany
{hassani,seidl}@cs.rwth-aachen.de
[2] Chair for Operating Systems, RWTH Aachen University, Aachen, Germany
ayman@lfbs.rwth-aachen.de

Abstract. Graphics Processing Units (GPUs) are used together with the CPU to accelerate a wide range of general purpose applications or scientific computations. The highly parallel architecture of the GPU consists of hundreds of cores optimized for parallel performance. Applications taking benefit of the GPU architecture have to be implemented according to the GPU parallel concept. An algorithm which follows a sequential work flow, has to be redesigned to achieve good performance on the GPU device. DenStream is a recent stream clustering algorithm that consists of two main parts. The online part summarizes data from the data stream, and builds micro clusters, while the offline part generates the final clustering using density-based clustering. In this work, we present a GPU-based efficient implementation of DenStream called (G-DenStream). G-DenStream is faster than DenStream, especially when the dimensionality of the streaming dataset increases, while keeping the quality of the reflected clustering as it is. The implementations in this work achieve palatalization of both online and offline parts and test the performance and the utilization on the GPU.

Keywords: GPGPU · Stream clustering over GPU · Parallel stream mining · Density-based stream clustering · G-DenStream

1 Introduction

Graphics processing units (GPUs), primarily designed for graphics application acceleration, are nowadays frequently used for general purpose computation. General-purpose computing on graphics processing units (GPGPU) gives the opportunity to use the high computation power and strong parallel architecture of the graphic coprocessor in combination with the *host* (the CPU). GPUs have become interesting for computing different tasks also outside the graphical domain, because of the good price-performance ratio, high fast memory transfer bandwidth and computation parallelism.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 441–453, 2014.
DOI: 10.1007/978-3-319-13186-3_40

There are a lot of solutions presented using GPUs in different scientific domains [6], like statistical physics, bioinformatics [7], segmentation, anti-virus software [12], audio signal processing etc., achieving substantial speed up against the homogenous CPU design and implementation.

Graphics Processing Units (GPUs) were primarily developed for a fast image rendering and a high performance computation required in computer graphics. As the image output in computer graphics is built on discrete grids, each grid element's output value needs to be computed as function of geometric and color arguments. Since no data dependency between the computations exists, all of them can be performed in parallel. This strategy of parallel computing achieves high performance building the graphical output. Another major goal of developing a separate device which is dedicated to graphical computations, is to reduce the CPU work load. Nowadays the GPU is a common part of the PC architecture, acting as a co-processor which is placed on a separate card and equipped with a dedicated memory. Using GPU devices for general purpose computations is motivated by the parallel architecture of the GPU, which offers high memory bandwidth and processing power at a low cost, and extends the host architecture using a powerful co-processor and high speed memory. In the data mining field, many problems deal with huge data sets in high time and space complexity. Data stream clustering processes evolving objects in the time data stream. The data stream delivers new data, which changes constantly the cluster distribution. A challenging aspect for data stream clustering algorithms is to process the new input data once only. An efficient memory management is required due to the fact that data streams are significantly bigger than the available memory on the most existent platforms. These issues were considered in a recent density-based stream clustering algorithm called DenStream [2], that lacks however to efficiency when handling huge data sets.

Motivated by the high performance of a data mining applications using the GPU, DenStream [2] is redesigned and integrated into a GPU scheduling environment. Our algorithm, called *G-DenStream* for a Gpu-based DenStream, presents a new parallel approach using the heterogeneous environment scheduling system for gaining speed ups in cases of clustering complex streaming high dimensional data. *G-DenStream* should not compromise the correctness and quality of the original DenStream, but should optimize it for GPU computing.

The remainder of this paper is organized as follows: Sect. 2 discusses some of the available related work. Section 3 explains the design concepts used for developing our G-DenStream algorithm, while Sect. 4 explains the two variants of the G-DenStream algorithm. Section 5 lists some of the experimental results we got after evaluating our algorithm. Section 6 concludes this paper with an outlook.

2 Related Work

In the related work we will talk about two main areas. First, we will explain in some details the DenStream [2] algorithm since it is the main algorithm that we build above while developing *G-DenStream*. Then, we talk about some available attempts to implement GPU-versions of known static or stream clustering algorithms.

2.1 DenStream Stream Clustering Algorithm

A data stream is a continuous data sequence without length limitation, containing (d, t)-*tuples* where d is a data record, and t a timestamp. DenStream is a data stream clustering approach, presented in [2]. DenStream uses exponentially fading function $f(t) = 2^{-\lambda t}, \lambda > 0$ to reduce the weight of the old data records in the time. Adjusting the value of λ, the importance of the historical data is set lower for higher λ value. Since the length of the data stream is theoretically infinite, not all arriving data points can be stored in the memory. In practice, even for finite data streams the stream data size can exceed the available memory.

DenStream uses micro clusters as summary representation for a group of close data points. The representation defines a spherical region in the data set with center c, radius $r < \epsilon$ and weight w. The micro clusters evolve with the time by the new data, changing w, c and r. At time t, the weight of the micro cluster is defined as the sum over the weights of the included points at the moment t. A threshold $\beta.minPts$ defines the type of the micro cluster. Micro clusters with $w \geq \beta.minPts$ are potential micro clusters and are grouped together to form the final clusters using DBSCAN. Micro clusters with $w < \beta.minPts$ are outlier micro clusters which are not dense enough at the current time. A very important property of the micro clusters is that they can be maintained incrementally. Adding new point to the micro cluster or fading by the fade function $f(\lambda)$ does not require recomputation of the micro cluster using the data of each single point in the micro cluster. When using this property, the stream data is processed only once, extracting the required features without saving each point in the memory for further use.

The incremental maintenance property of the micro clusters is achieved by storing the weighted linear and the weighted squared sum of the points in the micro cluster. For a micro cluster summarizing n points $p_1, p_2, \ldots, p_n$, with time stamps $T_1 < T_2 < \cdots < T_n$ and considering the fading function $f(\lambda)$, the linear sum at time t is $\overline{CF1} = \sum_{i=1}^{n} f(t - T_i)p_i$. The weighted squared sum for the same micro cluster is $\overline{CF2} = \sum_{i=1}^{n} f(t - T_i)p_i^2$.

Using $\overline{CF1}$, $\overline{CF2}$ and w as the features defining a micro cluster $mc = \{\overline{CF1}, \overline{CF2}, w\}$, a merge of a new point p is realized modifying mc to $mc = \{\overline{CF1} + p, \overline{CF2} + p^2, w + 1\}$. Fading out a micro cluster for an interval δt is performed using the transformation $mc = \{\overline{CF1}.2^{-\lambda \delta t}, \overline{CF2}.2^{-\lambda \delta t}, w.2^{-\lambda \delta t}\}$. The radius and center of a micro cluster, used in the DenStream algorithm are computed each time from the features $\overline{CF1}$, $\overline{CF2}$ and w. The DenStream algorithm allows a micro cluster to change its type or fade completely out and disappear. The period $Tp = \lceil \frac{1}{\lambda} \log(\frac{\beta\mu}{\beta\mu - 1}) \rceil$ determines how often a micro cluster should be checked whether it has faded out into being outlier. Thus, if some outlier micro cluster's weight w, at time $t \geq k.Tp$ for $k \geq 1$, is $w < \xi(t, t_0) = \frac{2^{-\lambda(t - t_0 + Tp))} - 1}{2^{-\lambda Tp} - 1}$ the micro cluster can be discarded.

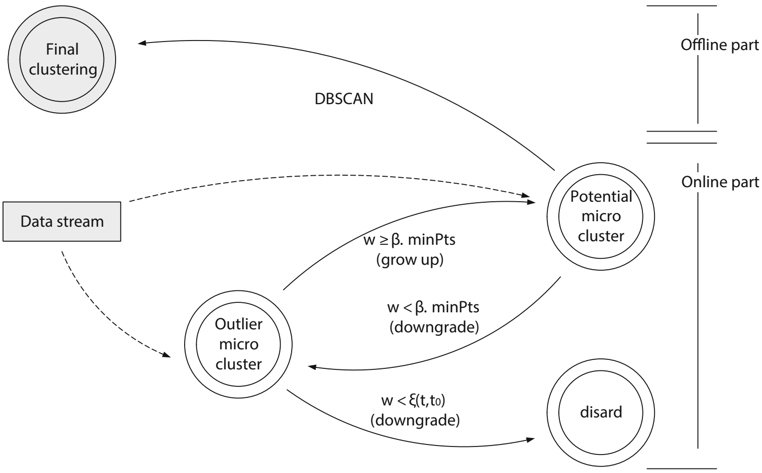


Fig. 1. DenStream and Micro cluster types

The DenStream clustering algorithm consists of two major parts, indicated also in Fig. 1:

Online part. This part of the algorithm is responsible for the micro cluster maintenance. The maintenance includes merging new data, creating new micro clusters and fading the existing micro clusters.

Offline part. In the offline part the final clustering is generated, clustering the potential micro clusters using DBSCAN. It is started upon a user request.

2.2 GPU Implementations of Clustering Algorithms

Different data mining approaches applied over the GPU have focused on performance gain using the GPU. One data stream solution using OpenCL is presented in [4]. The authors presented some methods for memory and work group size optimization for the clustering algorithm used, based on k-means. To save and reuse device memory the presented solution processes the problem domain in portions, optimizing the memory rakes size automatically for the hardware platform used. Different to [4], our algorithm *G-DenStream* deals with the more complex and the more accurate *density-based* stream clustering problem. *Static* clustering huge datasets, using GPUs and implemented with CUDA is discussed in [14]. The presented method uses asynchronous data transfers between the host and the device to perform the data copy, for a data block. Each data block is part of the data set which is too big to fit completely within the GPU's dedicated memory. Another works like [8,9], also focus on gaining speed ups against the CPU clustering version performing in parallel multiple distance computations required in the k-means clustering algorithm. Generally the most clustering applications using the GPU are based on *k*-means as a clustering algorithm.

In [5], memory access optimization of different OpenCL kernel are evaluated discussing k-means and GPGPU. In contrast to previous solutions, *G-DenStream* is dealing with streaming data. By definition, in DBSCAN [3], the clusters are collected point by point in levels, and expanding the edge of the cluster requires the whole computation of previews layer before proceeding to the next one. This feature is the key problem in the parallelization of DBSCAN. There are two interesting works which tried to overcome that are [1,11] by handling the problem in two different ways. Reference [1] starts multiple instances of DBSCAN and summarizes the results. The idea is presented in [1], using density chains defined as connected but not maximal dense regions. This method has relatively high memory consumption and uses functions with divergence program flow on the GPU. The strategy implemented in *G-DenStream* in the offline part follows a similar strategy to [11], when running only the time consuming part of DBSCAN. The similarity queries used for calculating the distances between the data points are executed in parallel on the GPU. This design decision ensures GPU task with low divergence and excellent parallelism, which yields a better exhausting of available resources. A content based similarity query GPU implementation is presented in [10]. This work shows efficient computation of the SQFD (Signature Quadratic Form Distance) using a GPU device managed by scheduling environment. The SQFD distance function is more complicated than the Euclidean distance used in the similarity queries for DenStream and shows good utilization and performance on the GPU.

3 Designing the *G-DenStream* Algorithm

The GPU computation follows a paradigm in which a massive data volume is processed in parallel for each data unit. An important constraint which cannot always be satisfied is that no dependency between the data units should exist to allow parallel processing. In this section, the main tasks used in the online and the offline parts are introduced. The left side of Fig. 2 shows the original DenStream algorithm (Sect. 2.1). On the right side, the corresponding G-DenStream tasks are marked. The diagram shows one iteration flow of the main program loop started for each point of the data stream.

3.1 The Online Part of G-DenStream

The online part of DenStream is executed for each new point arriving from the data stream. The part of the online phase of DenStream that collects the data to support the merge decision, is referred to as the *candidate* task in the computation. On the GPU device, all the distances between the new data point and the existing micro clusters' centers can be computed in an efficient parallel way. Each distance calculation is independent from the rest of the micro clusters. This allows starting as many instances of the distance computation kernel (threads) as micro clusters available, and running them in parallel on the GPU device. We extend the kernel function to compute also the new radius of a micro

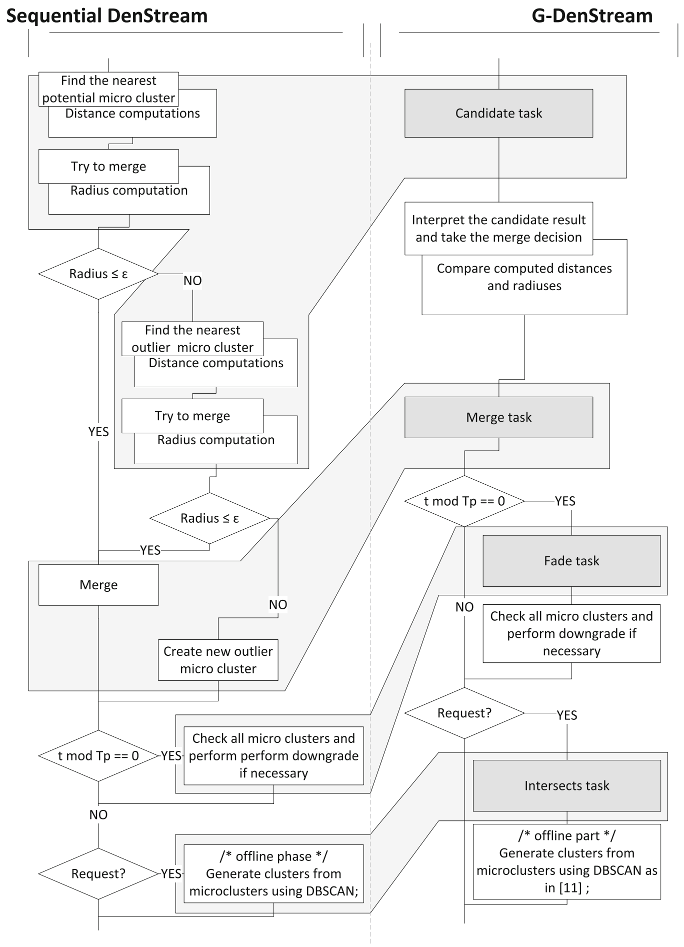

Sequential DenStream
G-DenStream
Find the nearest potential micro cluster
Distance computations
Try to merge
Radius computation
Radius ≤ ε
NO
YES
Find the nearest outlier micro cluster
Distance computations
Try to merge
Radius computation
Radius ≤ ε
YES
NO
Merge
Create new outlier micro cluster
t mod Tp == 0
YES
NO
Check all micro clusters and perform perform downgrade if necessary
Request?
YES
/* offline phase */ Generate clusters from microclusters using DBSCAN;
Candidate task
Interpret the candidate result and take the merge decision
Compare computed distances and radiuses
Merge task
t mod Tp == 0
YES
NO
Fade task
Check all micro clusters and perform downgrade if necessary
Request?
YES
Intersects task
/* offline part */ Generate clusters from microclusters using DBSCAN as in [11] ;

Fig. 2. From a sequential DenStream to a parallel G-DenStream design.

cluster caused by merging the new point, this kernel delivers all the data required for the merge decision in form of two vectors of length n, where n is the current number of micro clusters. To allow efficient memory accesses in the kernel function, all micro clusters are managed in one data structure, using a global ID in the kernel execution. The kernel function is started with a compute sub-task. The whole *candidate* task consists of a copy sub-task, transferring the new data to the device, the compute sub-task, and another copy sub-task that copies the results to the host. The online part is also responsible for fading out the old micro clusters, to give less importance exponential to old data. In practice, fading out is realized by multiplying the micro cluster's features $CF1$, $CF2$ and w with the fade factor described in Sect. 2.1. For the micro cluster that will include the current point, the multiplication is executed whenever the point is merged to it.

This ensures that the micro cluster fades out for the time period between the last and the current merge actions. This type of fade out is considered as implicit fade out, and is a part of the *merge* operation. However, only one micro cluster is updated with the current point. Thus, a separate *fade* task, independent from the *merge* is defined in the implementation part, to ensure that each micro cluster will fade out at most after a period Tp introduced in Sect. 2.1. The *fade* task also checks all the micro clusters for the minimum weight bound to keep them as potential or outlier micro clusters, following a special method of DenStream. We name this method the *downgrade* method in Line 11 of Algorithm 1. According to this method, potential micro clusters change into outlier, and outlier whose weight is below the lower bound can be deleted. The *fade* task can also be executed in parallel for all micro clusters. Unlike the *fade* and *candidate*, the *merge* operation is characterized by a divergent program flow and a concurrent memory access. For this reason, the *merge* part is not as optimally parallelized as the *fade* and *candidate* parts. The divergent program flow is caused by the decision part of the merge, which acts different to the cases of merging and creating new micro clusters. Even if some equal approach for both cases is assumed, that uses a dedicated data structure for this purpose, only one micro cluster can be updated without memory access conflicts or memory transfer overhead at a time. Resolving the divergence problem, results with a bad utilization or a high memory transfer overhead is caused. The other problem is the concurrent memory accesses generated by scanning the result vectors from the *candidate* task when searching for the minimum distance values of potential and outlier micro clusters.

By facing these problems, two different strategies of G-DenStream are presented and evaluated in the following section:

1. *Solution A* is to run the different task types on the suitable device, performing the *merge* on the CPU and the *candidate* and *fade* on the GPU.
2. *Solution B*, to start the *merge* also on the GPU. Although this task is not optimal, but we want to benefit from saving the memory transfers, by localizing as much calculation as possible on one device.

3.2 The Offline Part of G-DenStream

The offline part of DenStream algorithm runs a slightly modified version of the original DBSCAN over the current micro clusters. Shortly, we adopt in the offline pat a modified version of the implementation mentioned in [11].

4 Implementing the *G-DenStream* Algorithm

Two different implementations of *G-DenStream* are presented in this section. In both of them, the input data stream is read from file, where the sequence of lines in the file corresponds to the order of data points (represented by that line) in the stream. The program parameters are also stored in file, to allow saving the different test configurations. The stored parameters are the DenStream parameters λ, ϵ, β and $MinPts$, but also the period of output request *Rp*, page size for the memory paging and other implementation specific arguments. The third file required is the initialization file, which contains a subset of the data set for the initialization phase of *G-DenStream* according to the original algorithm. In both versions presented, the initialization is performed on the CPU.

The output is generated as a series of files, according to the run configuration. It includes the micro clusters, detected in the stream and the final clustering from the offline part. The user can adjust the period of output requests *Rp*. For example, when setting the request period of 1000, the output with the clustering will be generated every 1000 data records.

4.1 First Implementation: *Solution A*

The first version of *G-DenStream* utilizes the CPU and the GPU using asynchronous task starts and joins, performing parallel on both devices. The data structure is based on serialized feature vectors stored in the shared host memory and a copy on the GPU device memory. Having two redundant copies of the micro cluster data causes some overhead, but is necessary, since one part of the algorithm runs on the host and another part on the GPU device. *Solution A* (cf. Algorithm 1) keeps this overhead as small as possible, by performing incremental

Algorithm 1. Pseudocode of the host program of *Solution A*

```
Data: $Rp, DataStream, \lambda, \epsilon, \beta, MinPts$
Result: Clustering
initialization();
forall the $p \in DataStream$ do
    if request(Rp) then Request.reset; Request.start;
    CandidateTask($p$).reset;
    CandidateTask($p$).start;
    CandidateTask($p$).join;
    if fadePhase($\lambda$) then fade.reset; fade.start;
    Scan(CandidateTaskResult);
    if fadePhase($\lambda$) then fade.join;
    merge($p$);
    downgradeMCs($\beta, MinPts$);
    if request(Rp) then
        Request.join;
        Request.DBSCAN; Request.output;
    end
    if MicroClusters >= MemRange then expandMemRange();
end
```

updates only over part of the data copy and running the big memory transfer tasks in background in parallel to other computations.

4.2 Second Implementation: *Solution B*

There are three major problems in *Solution A*: (1) The high memory transfer volume, (2) the high number of starting tasks, which causes a big scheduling time overhead and (3) the processing of only one point per iteration and time unit in the main program loop. There is a relationship between the high number of tasks and the manner of data stream processing. Since only one point is merged per program loop, the application needs a high number of loops to manage the data stream. For each iteration several tasks are used, which will have in consequence a task scheduling overhead.

Using the three tasks from the online and offline part of *Solution A* of *G-DenStream*, one of the following four task compositions is started in each host program loop within *Solution B* (cf. Algorithm 2):

Algorithm 2. Pseudocode of the host program of *Solution B*

```
Data: $Rp, DataStream, \lambda, \epsilon, \beta, MinPts, pps$
Result: Clustering
initialization();
forall the $p \in DataStream$ do
    build packageOfPoints();
    if packageOfPoints.size() == pps then
        if request(Rp) then
            if fadePhase($\lambda$) then
                C4.run();
            else
                C3.run();
            end
            Request.DBSCAN();
            Request.output();
        else
            if fadePhase($\lambda$) then
                C2.run();
            else
                C1.run();
            end
        end
    end
    if MicroClusters >= MemRange then expandMemRange();
end
```

C1: Start, join and reset the CM-task (candidate and merge). In this case the new data is transferred to the device using a copy sub-task, followed by a compute sub-task and another copy sub-task, returning the memory state of the device.

C2: Start, join and reset the FCM-task (fade, candidate and merge), also consisting of three sub-tasks: copy, compute and copy.

C3: Start, join and reset the CM-task, followed by start, join and reset of the task supporting the offline phase of DenStream.

C4: Start, join and reset the FCM-task, followed by start, join and reset of the task supporting the offline phase of DenStream.

5 Experimental Evaluation

To compare both *G-DenStream* implementations presented in *Solution A* and *Solution B* with the original DenStream algorithm, DenStream was also implemented over the CPU only. The clustering results from the different implementations are verified using synthetic and real data sets. Since all three implemented algorithms deliver identical results, we can concentrate on the performance evaluation and characteristics of the different GPU variants.

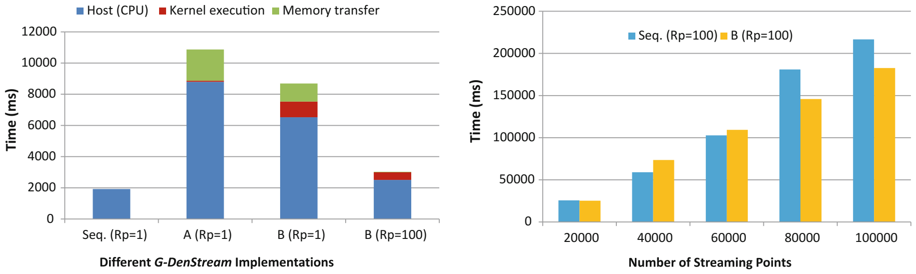
5.1 Hardware Environment

All tests are performed on a Fedora 17 Linux PC with a Quad-Core-CPU Intel(R) Core(TM) i5-3550 CPU @ 3.30 GHz and 4 GB RAM. The test platform is equipped with an AMD Radeon HD 7870 GPU, with the following technical specifications: 1000 MHz Engine Clock, 2 GB GDDR5 Memory, 1200 MHz Memory Clock (4.8 Gbps GDDR5), 153.6 GB/s maximal memory bandwidth, 2.56 TFLOPS in Single Precision, 20 Compute Units (1280 Stream Processors), 256-bit GDDR5 memory interface, PCI Express 3.0 × 16 bus interface and an OpenCL(TM) 1.2 support.

5.2 Datasets

One synthetic and another real data set are used to test the quality and performance of the clustering algorithms implemented in Solutions A and B, compared with the CPU implementation of DenStream. The synthetic data set is from [13] and includes 31 synthetically generated clusters. The data in each cluster is normally distributed around the center of the cluster. The data set contains 3100 two dimensional points.

Further performance analysis of *G-DenStream* is tested on a real data set. The Physiological Data Modeling Contest at ICML 2004 data set is a collection of activities which was collected by test subjects with wearable sensors over several months. The dataset consists of 720792 data objects, each data object has 15 attributes and consists of 55 different labels for the activities and one additional label if no activity was recorded. We picked 9 numerical attributes.



(a) (b)

Fig. 3. (a) Running time distribution for the different implementations of *G-DenStream* using the synthetic dataset (seq. is the sequential DenStream), (b) *Solution B* compared to the sequential DenStream using the ICML data set

5.3 Experimental Results

All the time measurements include the total execution time of DenStream, containing the initialization phase, stream processing and the request response every *Rp* points.

Figure 3(a) compares the sequential DenStream with both *G-DenStream* variants *A* and *B*, using the synthetic data set. *Solution A* runs significantly slower than all other designs, heavily loaded by the host side part of the algorithm and by the memory transfers. The host side part of the algorithm starts multiple tasks per merged point and causes a time overhead, which cannot be compensated by the optimal kernels running only 71 ms of the whole program running time. Combining multiple kernels to reduce the task starts in *Solution B*, shows positive impact for the summarized program running time, but a negative impact over the total kernel time. This effect is caused by the *merge* action which is sequentially executed in one instance of the kernel execution. The thread performing the *merge* runs longer then the other kernel instances and slows down the synchronization for the next point processing.

The next experimental evaluation in Fig. 3(b) is performed with the high dimensional data set used in the Physiological Data Modeling Contest at ICML 2004. *Solution B*, executed with $Rp = 100$ performs faster than the sequential DenStream. For such a big data set, the offline part computation for a high dimensional data, generates a significant CPU load in the sequential version of DenStream. The same computation runs efficiently on the GPU using *G-DenStream*, by reducing the CPU load and running in parallel with other GPU tasks, and thus exhausting the underlying resources.

6 Conclusion and Future Directions

In this paper, we presented two variant GPU-based implementations of the density based clustering algorithm *G-DenStream*. DenStream was redesigned to meet the characteristics and special design requirements of the parallel GPU architecture. The fade, candidate and intersects task were defined and implemented in a parallel way for the GPU. *Solution A* differs from *B* by the *merge* task. *Solution B* performed better including the *merge* part of DenStream in the GPU kernel, and saves task starts and memory transfers between the host and the device.

In the future, we would like to further optimize *G-DenStream* to additionally exhaust the available resources. One optimization method for *Solution B* could be the transformation of the *merge* routine to perform also in parallel. This change will allow the merging of more than one point by parallel threads, instead of one point per compute task and sequentially.

Acknowledgments. This work has been partially supported by the UMIC Research Centre, RWTH Aachen University, Germany.

References

1. Böhm, C., Noll, R., Plant, C., Wackersreuther, B.: Density-based clustering using graphics processors. In: Proceedings of CIKM'09, pp. 661–670 (2009)
2. Cao, F., Ester, M., Qian, W., Zhou, A.: Density-based clustering over an evolving data stream with noise. In: Proceedings of SDM'06, pp. 328–339 (2006)
3. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD'06, pp. 226–231 (2006)
4. Fang, J., Varbanescu, A.L., Sips, H.: An auto-tuning solution to data streams clustering in opencl. In: Proceedings of IEEE International Conference on Computational Science and Engineering, pp. 587–594 (2011)
5. Gunarathne, T., Salpitikorala, B., Chauhan, A., Fox, G.: Iterative statistical kernels on contemporary GPUs. Int. J. Comput. Sci. Eng. **8**, 58–77 (2013)
6. McCool, M.: Signal processing and general-purpose computing and GPUs [exploratory dsp]. IEEE Signal Process. Mag. **24**, 109–114 (2007)
7. Schatz, M., Trapnell, C., Delcher, A., Varshney, A.: High-throughput sequence alignment using graphics processing units. BMC Bioinform. **8**, 474 (2007)
8. Shalom, S.A., Dash, M., Tue, M.: Efficient k-means clustering using accelerated graphics processors. In: Proceedings of DaWaK'08, pp. 166–175 (2008)
9. Takizawa, H., Kobayashi, H.: Hierarchical parallel processing of large scale data clustering on a PC cluster with GPU co-processing. J. Supercomput. **36**, 219–234 (2006)
10. Tarakji, A., Hassani, M., Lankes, S., Seidl, T.: Using a multitasking GPU environment for content-based similarity measures of big data. In: Murgante, B., Misra, S., Carlini, M., Torre, C.M., Nguyen, H.-Q., Taniar, D., Apduhan, B.O., Gervasi, O. (eds.) ICCSA 2013, Part V. LNCS, vol. 7975, pp. 181–196. Springer, Heidelberg (2013)
11. Thapa, R., Trefftz, C., Wolffe, G.: Memory-efficient implementation of a graphics processor-based cluster detection algorithm for large spatial databases. In: IEEE International Conference on Electro/Information Technology (EIT'10), pp. 1–5 (2010)
12. Vasiliadis, G., Antonatos, S., Polychronakis, M., Markatos, E.P., Ioannidis, S.: Gnort: high performance network intrusion detection using graphics processors. In: Lippmann, R., Kirda, E., Trachtenberg, A. (eds.) RAID 2008. LNCS, vol. 5230, pp. 116–134. Springer, Heidelberg (2008)

13. Veenman, C., Reinders, M.J.T., Backer, E.: A maximum variance cluster algorithm. IEEE Pattern Anal. Mach. Intell. **24**, 1273–1280 (2002)
14. Wu, R., Zhang, B., Hsu, M.: Clustering billions of data points using GPUs. In: Proceedings of the Combined Workshops on UnConventional High Performance Computing Workshop Plus Memory Access Workshop, pp. 1–6 (2009)

Algorithms for Large-Scale Information Processing in Knowledge Discovery

Three-way Indexing ZDDs for Large-Scale Sparse Datasets

Hiroshi Aoki[1(✉)], Takahisa Toda[2], and Shin-ichi Minato[1,3]

[1] Graduate School of Information Science and Technology,
Hokkaido University, Sapporo, Japan
{kioa,minato}@ist.hokudai.ac.jp

[2] Graduate School of Information Systems,
University of Electro-Communications, Chofu, Japan
toda.takahisa@gmail.com

[3] ERATO MINATO Discrete Structure Manipulation System Project,
Japan Science and Technology Agency,
Hokkaido University, Sapporo, Japan

Abstract. Zero-suppressed decision diagrams (ZDDs) are a data structure for representing combinations over item sets. They have been applied to many areas such as data mining. When ZDDs represent large-scale sparse datasets, they tend to obtain an unbalanced form, which results performance degradation. In this paper, we propose a new data structure three-way indexing ZDD, as a variant of ZDDs. We furthermore present algorithms to convert between three-way indexing ZDDs and ordinary ZDDs. Experimental results show the effectiveness of our data structure and algorithms.

Keywords: ZDD · Zero-suppressed binary decision diagram · Ternary search tree · Membership query

1 Introduction

Combinations over item sets are a fundamental notion. Computational problems concerning combinations naturally arise in many fields of computer science. Recently, finding useful information from large-scale datasets has been extensively studied in data mining research [1]. Some of these datasets can be regarded as sets of combinations, such as transaction databases. To solve combinatorial problems efficiently, it would be helpful to have an efficient representation for sets of combinations.

A *zero-suppressed binary decision diagram* (ZDD in short) indexes and stores combinations as a directed graph [2]. A ZDD is a compressed representation in the sense that it can be obtained from a binary search tree by reducing irrelevant nodes and sharing equivalent subgraphs. This makes it space-efficient. Furthermore, there are efficient operations to manipulate sets of combinations over ZDDs; among the operations efficient on ZDDs are union, intersection, and

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 457–469, 2014.
DOI: 10.1007/978-3-319-13186-3_41

set difference. As on binary search trees, membership queries can be answered by traversing a path from the root.

ZDDs have many applications. In many data mining tasks, generating interesting patterns such as frequent itemsets from datasets plays an essential role. Minato et al. proposed a technique to output such generated patterns as a ZDD [3] in a compact form. Minato further proposed various algorithms to extract interesting structural information hidden in the output ZDD [4–6]. Such algorithms are also important in practical data mining applications. See [7–9] for other applications of ZDDs.

ZDDs can efficiently represent sets of combinations, but ZDDs do not always achieve very good performance in practice, particularly for large-scale sparse datasets. If we represent such datasets using ZDDs, the height of a ZDD grows as large as the total number of items, and the depth of recursive operations also becomes very large. Thus, ZDD operations that depend on path lengths are usually not very efficient for large-scale sparse datasets. For this, Minato proposed Z-skip-links [10], a technique to reduce the depth of recursion by adding one link per ZDD node. Since large-scale sparse datasets are important in practice, it is important to develop techniques to compensate for the drawbacks of ZDDs.

In this paper, we present an alternative solution to the path length problem described above. We propose a new data structure, *three-way indexing ZDD* (3-way ZDD), which is a variant of a ZDD. This data structure is a useful combination of the *ternary search tree* considered in [11] and the ZDD-based compression technique. As mentioned in [11], a node in a ternary search tree represents a subset of vectors by a partitioning value and three vector references: one to lesser elements, one to greater elements (as in a binary search tree), and one to equal elements; equal elements are then processed on later fields (as in tries). For performance reasons, the partitioning value is chosen as either randomly or from an estimated median. Our idea is to compute the true median, for which combinations are uniquely represented by a ternary search tree, and then to reduce nodes by sharing all equivalent subgraphs in the ternary search tree. Note that computing the true median is necessary because otherwise it would be possible to have distinct subgraphs representing the same set of combinations; such subgraphs cannot be shared. We refer to the graph obtained from a ternary search tree by use of the true median as a 3-way ZDD. We present algorithms to convert between 3-way ZDDs and ordinary ZDDs. We expect that these algorithms will be useful because 3-way ZDDs and ordinary ZDDs have complementary strengths; this is discussed in Sect. 4. In this paper, we experimentally show the effectiveness of our data structure and algorithms.

The paper is organized as follows. In Sect. 2, we introduce basic definitions and results of ZDDs. In Sect. 3, we propose and define three-way indexing ZDDs. In Sect. 4, we discuss advantages and disadvantages of 3-way ZDDs and then present algorithms to convert between 3-way ZDDs and ordinary ZDDs. In Sect. 5, we present experimental results. Section 6 concludes the paper.

2 Preliminaries

2.1 ZDDs

A ZDD [2] is a graph-representation of a set of combinations. A *combination* is a subset of items. Figure 1 shows an example of a ZDD. The node at the top is called the *root*. Each internal node has the three fields V, LO, and HI. The field V holds an item, which for simplicity we suppose is denoted by a positive number. The fields LO and HI point to other nodes, which are called the LO and HI *children*, respectively. The arc to a LO child is called a LO *arc* and illustrated by a dashed arrow; the arc to a HI child is called a HI *arc* and illustrated by a solid arrow. There are only two terminal nodes $\top$ and $\bot$.

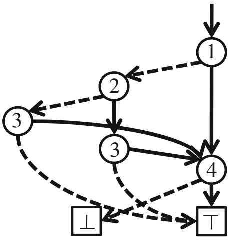
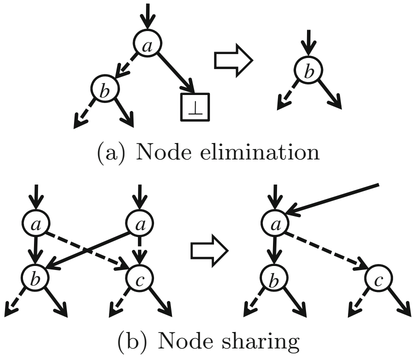


Fig. 1. The ZDD for the set family $\{\emptyset, \{1,4\}, \{2\}, \{2,3,4\} \{3,4\}\}$.



(a) Node elimination

(b) Node sharing

Fig. 2. Reduction rules on ZDDs.

For efficient compression, ZDDs must satisfy the following two conditions. They must be *ordered*: if a node u points to a node v, then $\mathrm{V}(u) < \mathrm{V}(v)$. They must be *reduced*: the ZDD must be invariant under the following two reduction operations cannot be applied.

1. For each internal node u whose HI arc points to $\bot$, redirect all the incoming arcs of u to the LO child, and then eliminate u (Fig. 2(a)).
2. For any nodes u and v, if the subgraphs rooted by u and v are equivalent, then share the two subgraphs (Fig. 2(b)).

We can understand ZDDs as follows. Given a ZDD, each path from the root to $\top$ corresponds to a combination in such a way that an item k is included in the combination if a path contains the HI arc of a node with label k; otherwise, k is excluded from the second path. For example, in Fig. 1, the paths ①--→ ②--→ ③--→ $\top$, ①→ ④→ $\top$, ①--→ ②→ ③--→ $\top$, ①--→ ②→ ③→ ④→ $\top$ and ①--→ ②--→ ③→ ④→ $\top$ correspond to $\emptyset$, $\{1,4\}$, $\{2\}$, $\{2,3,4\}$ and $\{3,4\}$, respectively.

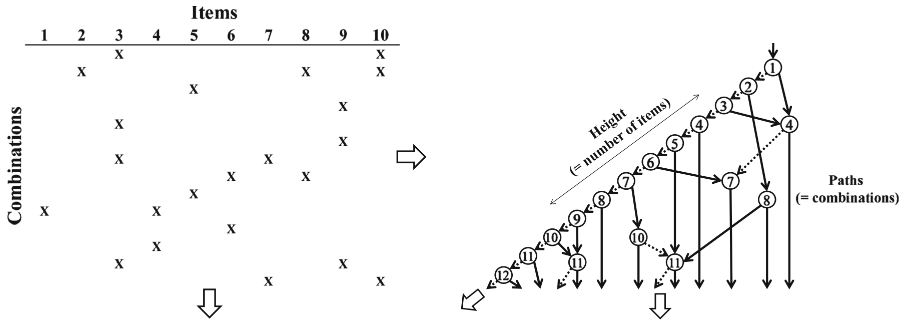
Note that although the node ③ does not appear in the second path, the node elimination rule implies that the HI arc of ③ points to ⊥, and thus 3 is excluded.

It is known (see for example [2,12]) that for any set of items V, every set of combinations over V corresponds to a unique ZDD if the order of items in V is fixed. ZDD nodes are maintained by a hash table, called a *uniquetable*, so that for a triple (k, l, h) of a node label and two ZDD nodes, there is a unique ZDD node p with $\mathrm{V}(p) = k$, $\mathrm{LO}(p) = l$, and $\mathrm{HI}(p) = h$. Given a triple (k, l, h), the function ZDD_UNIQUE returns an associated node in the uniquetable if such a node exists. Otherwise, it creates a new node p such that $\mathrm{V}(p) = k$, $\mathrm{LO}(p) = l$, and $\mathrm{HI}(p) = h$; p is then registered in the uniquetable and returned. A uniquetable guarantees that two nodes are different if and only if the subgraphs rooted by them represent different sets of combinations. Thus, for example, equivalence checking of sets of combinations can be done in constant time.

We here introduce notation and terminology, and make a remark. The *size* of a ZDD u is the number of nodes in the ZDD, denoted by $|u|$. We identify a given node with the ZDD rooted by the node. For any node in a ZDD, the subgraph rooted by the node is itself a ZDD.

2.2 ZDDs Representing Large-Scale Sparse Datasets

ZDDs can efficiently represent sets of combinations in a compressed form, but ZDDs do not always perform well in practice. On ZDDs representing That is, when ZDDs represent large-scale *sparse* datasets, operations that depend on path length, such as membership query, are efficient. This paper considers datasets that contain a massive number of sparse combinations over a large base set. That is, such datasets are large in the sense that a very large number of combinations are represented and sparse in the sense that each such combination consists of a few items from a base set containing thousands or more items; this is illustrated in Fig. 3(a).



Combinations	1	2	3	4	5	6	7	8	9	10
			x							x
		x						x		x
					x					
									x	
			x							
									x	
			x				x			
						x		x		
					x					
	x			x						
						x				
				x						
			x						x	
							x			x

(a) Large-scale sparse dataset

(b) Corresponding ZDD

Fig. 3. Large-scale sparse dataset and corresponding ZDD.

Figure 3(b) shows a ZDD corresponding to the matrix in Fig. 3(a). Each row of the matrix corresponds to a path from the root to ⊤ of the ZDD. In such a ZDD, the number of LO arcs in a path tends to be much larger than the number of HI arcs. Thus, paths representing combinations of a few items may be many orders of magnitude longer than the number of items in the combinations. In such unbalanced ZDDs, operations that depend on path length, such as membership query would become inefficient because a large number of irrelevant LO arcs must be traversed in the execution of the operation. In the worst case, such an operation can require time proportional to the total number of items in the base set.

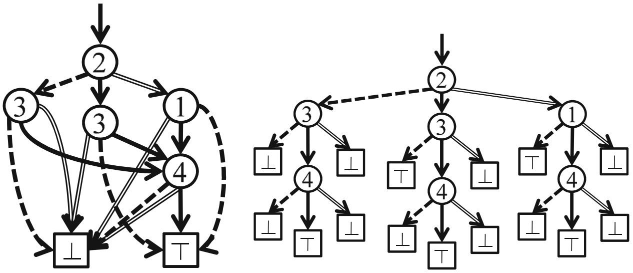
This paper proposes a new data structure that avoids this situation; the data structure is a variant of ZDD.

3 3-way ZDDs

We propose a new data structure the *3-way ZDD*. Figure 4(a) shows an example of a 3-way ZDD. Like ZDDs, each path of a 3-way ZDD corresponds to a combination over an item set. In a 3-way ZDD, each node has three children. We can understand 3-way ZDDs in the following way. In Fig. 4(a), the paths ②⇒ ①⇢ ⊤, ②⇒ ①→ ④→ ⊤, ②→ ③⇢ ⊤, ②→ ③→ ④→ ⊤ and ②⇢ ③→ ④→ ⊤ correspond to $\emptyset$, $\{1,4\}$, $\{2\}$, $\{2,3,4\}$ and $\{3,4\}$, respectively.

Three-way indexing ZDDs are derived from *ternary search trees* as follows. A ternary search tree is a data structure for sets of strings (see Fig. 4(b)) and satisfies the following conditions.

- Each internal node has the four fields V, LO, HI, and RO. The field V holds a character. The fields LO, HI and RO point to other nodes, which are called the LO, HI and RO *children*, respectively. LO and RO arcs mean that the character in V does not occur and a HI arc means that the character in V occurs.



(a) A 3-way ZDD (b) A ternary search tree

Fig. 4. The 3-way ZDD and ternary search tree for the family of sets $\{\emptyset, \{1,4\}, \{2\}, \{2,3,4\} \{3,4\}\}$.

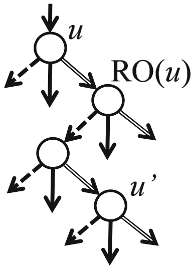


Fig. 5. Condition on order of node labels: $\mathrm{V}(u') < \mathrm{V}(u)$.

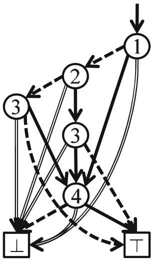


Fig. 6. A three-way ZDD not satisfying the balancing condition.

- As shown in Fig. 5, for each node u, every node u' reachable by following LO and RO arcs from the RO child of u has a character less than the character of u: $\mathrm{V}(u') < \mathrm{V}(u)$. This condition includes that $\mathrm{V}(\mathrm{RO}(u)) < \mathrm{V}(u)$. Similarly, every node u'' reachable by following LO and RO arcs from the LO child of u has a character larger than the character of u: $\mathrm{V}(u'') > \mathrm{V}(u)$. This includes that $\mathrm{V}(u) < \mathrm{V}(\mathrm{LO}(u))$.
- The HI arc a node must not point to $\perp$.

Since this paper treats sets of combinations instead of sets of strings, we assume that each node u is indexed by a natural number $\mathrm{V}(u)$ and that the following condition is satisfied: for each node u, $\mathrm{V}(u) < \mathrm{V}(\mathrm{HI}(u))$. Three-way indexing ZDDs can be obtained from ternary search trees by sharing all equivalent subgraphs.

As mentioned in [11], ternary search trees are not space-efficient compared to hash tables, and it is necessary to mitigate this drawback. For this, our basic idea is to reduce nodes by sharing equivalent subgraphs. However, this idea is not without its own difficulty. A set of combinations can be represented by multiple inequivalent 3-way ZDDs, as shown in Fig. 6. Figures 6 and 4(a) represent the same set of combinations. It is possible that in a 3-way ZDD, there may be distinct subgraphs that nevertheless represent the same set of combinations; being distinct, such subgraphs cannot be shared. Even worse, the problem of performance degradation on sparse datasets still remains. To solve these problems, we use *balanced* 3-way ZDDs, which satisfy the following condition: for each node u, it holds that $\mathrm{W}(\mathrm{LO}(u)) < \lceil \mathrm{W}(u)/2 \rceil \leq \mathrm{W}(\mathrm{LO}(u)) + \mathrm{W}(\mathrm{HI}(u))$, where $\mathrm{W}(u)$ denotes the number of paths from u to $\top$, which is called the *weight* of u. Unless otherwise mentioned, by 3-way ZDDs we will mean balanced 3-way ZDDs.

Proposition 31. *There is a one-to-one correspondence between families of subsets of* $\mathbb{N}$ *and balanced 3-way ZDDs.*

To keep uniqueness of nodes efficiently as ZDDs do, 3-way ZDD nodes are also maintained by in a hash table. Given a quad (k, l, h, r), the function TDD_UNIQUE returns the associated node in the uniquetable if exists; otherwise, the function creates a new node p such that $\mathrm{V}(p) = k$, $\mathrm{LO}(p) = l$, $\mathrm{HI}(p) = h$, and $\mathrm{RO}(p) = r$; p is registered to the uniquetable and returned.

We here introduce notation and terminology, and make a remark. The *size* of a 3-way ZDD u is the number of nodes in the 3-way ZDD; the size of u is denoted $|u|$. We identify each node with the 3-way ZDD rooted by that node. In any node of a 3-way ZDD, the subgraph rooted by that node is itself a 3-way ZDD.

Theorem 32. *A membership query on a 3-way ZDD representing n combinations that hold k items requires at most $\lfloor \lg n \rfloor + k$ node label comparisons, and this is optimal.*

Theorem 32 is introduced from comparison times of membership queries on a perfectly balanced ternary search tree shown in [11,13] because a membership query on a 3-way ZDD obviously need the same number of node label comparisons on a perfectly balanced ternary search tree.

4 Algorithms to Convert Between Ordinary ZDDs and 3-way ZDDs

4.1 Advantages and Disadvantages of 3-way ZDDs

Three-way indexing ZDDs and ordinary ZDDs complement each other. As mentioned in Sect. 2.1, ZDDs both efficiently compress sets of combinations and have *dynamic operations*, such as union, intersection, and set difference, to construct new ZDDs from input ZDDs. However, when ZDDs are unbalanced, it is not efficient to process membership and pattern match queries. We refer to these queries as *static operations*, because such operations do not change their input. In contrast to static operations on ordinary ZDDs, static operations can guarantee efficiency, because the balancing condition prevents 3-way ZDDs from becoming unbalanced. However, the tradeoff is that it may be difficult to efficiently execute dynamic operations on 3-way ZDDs because of this same balancing condition. Because ordinary ZDDs and 3-way ZDDs are useful in different contexts, we here present algorithms to efficiently convert between ordinary ZDDs and 3-way ZDDs.

4.2 Conversion from Ordinary ZDDs to 3-way ZDDs

Figure 7 shows the function Z2T to convert ZDDs to 3-way ZDDs. As shown in Fig. 9, the main idea of this algorithm is to recursively divide an input ZDD into three parts, which will correspond to the three children of the output 3-way ZDD.

First, the function FIND_PIVOT computes the ZDD node p at which an input ZDD z should be divided into three parts (see Fig. 8). We then obtain the following three ZDDs: the subgraphs rooted by the LO and RO children of p, and the ZDD $z \setminus p$, obtained from z by removing p. These ZDDs are recursively converted to 3-way ZDDs. We finally obtain an output node with the resultant 3-way ZDDs as its children.

```
function Z2T($z$)
    if $z = \top$ then
        return $\top$;
    end if
    if $z = \bot$ then
        return $\bot$;
    end if
    $p \leftarrow \mathrm{FIND\_PIVOT}\,(z)$;
    $z_0 \leftarrow \mathrm{LO}\,(p)$; $z_1 \leftarrow \mathrm{HI}\,(p)$; $z_2 \leftarrow z \setminus p$;
    $t_0 \leftarrow \mathrm{Z2T}\,(z_0)$; $t_1 \leftarrow \mathrm{Z2T}\,(z_1)$; $t_2 \leftarrow \mathrm{Z2T}\,(z_2)$;
    $t \leftarrow \mathrm{TDD\_UNIQUE}\,(\mathrm{V}\,(p)\,, t_0, t_1, t_2)$;
    return $t$;
end function
```

Fig. 7. Convert ordinary ZDDs to 3-way ZDDs.

```
function FIND_PIVOT($z$)
    $p \leftarrow z$
    while $\neg(\mathrm{W}\,(\mathrm{LO}\,(p)) < \lceil \mathrm{W}\,(z)\,/2 \rceil \leq \mathrm{W}\,(p))$ do
        $p \leftarrow \mathrm{LO}\,(p)$;
    end while
    return $p$;
end function
```

Fig. 8. Compute the ZDD node p at which an input ZDD z should be divided into three parts.

We remark that the weights of nodes in an input ZDD can be computed in time proportional to the size of the input ZDD [12]. We therefore perform this preprocessing before executing Z2T.

Theorem 41. *The algorithm* Z2T *can be computed in time proportional to* $O(|z|n\log n)$*, where* n *is the total number of distinct items that appear in the input ZDD.*

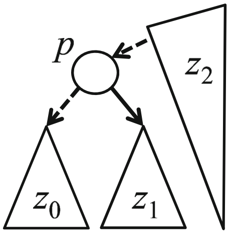


Fig. 9. Recursively three-partitioning a ZDD by Z2T.

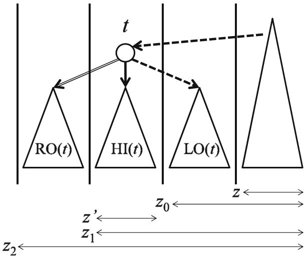


Fig. 10. Bottom-up construction of a ZDD by T2Z.

4.3 Conversions from 3-way ZDDs to Ordinary ZDDs

Figure 11 shows the function T2Z to convert 3-way ZDDs to ordinary ZDDs. At each recursive call, this function receives a 3-way ZDD and an intermediate-result ZDD. This function should be initially called with $z = \bot$. As shown in Fig. 10, the algorithm constructs a ZDD in a bottom up fashion in the order z_0, z_1, and z_2. Because Fig. 11 is a straightforward application of a standard technique, we omit a detailed explanation.

```
function T2Z($t, z$)
    if $t = \top$ then
        return $\top$;
    end if
    if $t = \bot$ then
        return $\bot$;
    end if
    $z_0 \leftarrow \mathrm{T2Z}(\mathrm{LO}(t), z)$;
    $z' \leftarrow \mathrm{T2Z}(\mathrm{HI}(t), \bot)$;
    $z_1 \leftarrow \mathrm{ZDD\_UNIQUE}(\mathrm{V}(t), z_0, z')$;
    $z_2 \leftarrow \mathrm{T2Z}(\mathrm{RO}(t), z_1)$;
    return $z_2$;
end function
```

Fig. 11. Converting 3-way ZDDs to ordinary ZDDs.

Theorem 42. *The algorithm* T2Z *can be computed in time proportional to $O(|t|n)$, where n is the total number of distinct items that appear in the input 3-way ZDD.*

5 Experimental Results

Implementation and Environment. We implemented 3-way ZDDs and conversion algorithms in C. We used the BDD Package SAPPORO-Edition-1.0 developed by Minato. In this library, ordinary ZDDs are available and basic operations for ZDDs are provided. All experiments were performed on a 2.67 GHz Xeon®E7-8837 with 1.5 TB of RAM, running SUSE Linux Enterprise Server 11. We compiled our code with version 4.3.4 of the gcc compiler.

Problem Instances. We used total 25 datasets, which were classified into the five types. The first dataset type consisted of randomly generated datasets. The other datasets represented maximal frequent sets, which are important in frequent itemset mining.

1. random(n): each row in a dataset contains 10 distinct items on average, uniformly randomly selected from $\{1, \ldots, n\}$. We generated these datasets by using the pseudorandom number generator Mersenne Twister, which is provided in GSL-1.11.
2. mf-accidents(n), mf-connect(n), mf-kosarak(n), mf-pumsb(n): each dataset corresponds to a maximal frequent itemset[1] with support threshold n for the datasets "accidents", "connect", "kosarak", and "pumsb", respectively. These datasets are standard experimental data in data mining research. We used LCM ver. 5.3 to generate maximal frequent itemsets.

[1] A *frequent itemset* with support threshold n is an itemset X such that the number of rows containing all items in X is more than or equal to n. A frequent itemset is *maximal* if it is not contained in any other frequent itemset with the same support threshold.

Comparison of Data Structures. We compared the *sizes* of the following 4 representations for sets of combinations: *a list of lists*, a ternary search tree, a three-way ZDD, and an ordinary ZDD. Here, size means the number of nodes, and a list of lists is a linked list such that each node points to a linked list representation of the items in a row of a dataset. A list of lists can be considered an uncompressed representation because the entries in a dataset correspond in a one-to-one way to the item nodes in a list of lists. A ternary search tree can be considered a compressed representation of a list of lists, and a 3-way ZDD can be considered a compressed ternary search tree because a ternary search tree can be obtained by sharing common list prefixes and a 3-way ZDD can be obtained by also sharing equivalent subgraphs.

The left panel in Fig. 12 shows a scatter plot in which each point represents a dataset. The locations are determined by the ratio of the size of a ternary search tree to the size of a list of lists (the horizontal coordinate) and the ratio of the size of a 3-way ZDD to the size of a ternary search tree (the vertical coordinate). Random datasets receive almost no benefit from compression, which can be seen by clustering near the upper right corner. The other datasets achieve good compression efficiency. From the right panel, we can observe that the size of a 3-way ZDD almost equals that of an ordinary ZDD. This implies that when converting between 3-way ZDDs and ordinary ZDDs, large changes data representation size are not a concern.

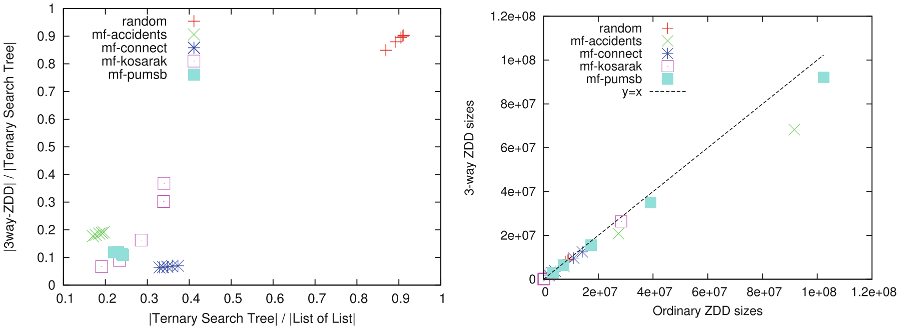


Fig. 12. Comparison of sizes in various representations: |LIL| is the size of a list of lists; |TST| is the size of a ternary search tree; |TDD| is the size of a 3-way ZDD.

Comparison of Membership Query Processing between Three-way ZDDs and Ordinary ZDDs. We compared the time required to process membership queries in 3-way ZDDs and ordinary ZDDs. The experiment for 3-way ZDDs is described below. From each dataset, we randomly selected 1,000,000 rows with repetition; we then constructed the 3-way ZDD. Next, for each such row, queried the 3-way ZDD for membership, which traversed the corresponding path. We measured the time used in traversal only. The same experiment was conducted for ordinary ZDDs. We can see that 3-way ZDDs effectively prevent degradation in membership query performance (Fig. 13).

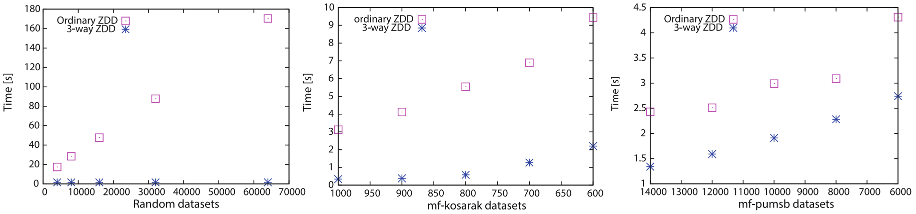


Fig. 13. Comparison of the total time for membership queries in 3-way ZDDs and ordinary ZDDs.

Performance of Conversion Algorithms. Theorems 41 and 42 state that the performance of the conversion algorithms Z2T and T2Z depends on the sizes of the input ZDD or 3-way ZDD, respectively. The experimental results in Fig. 14 support these theoretical results. Each point represents a dataset by its input size (horizontal coordinate) and running time (vertical coordinate). We can observe that the points nearly form a straight line. Because input sizes are widely distributed, the horizontal axis and the vertical axis are logarithmic in scale. Instances of the same type have the same color and shape.

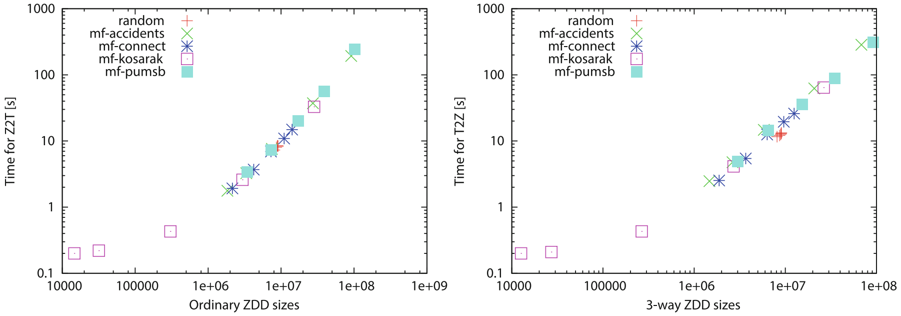


Fig. 14. Performance of conversion algorithms.

We furthermore compared the running time between our conversion algorithms and naive methods. A naive method to convert ordinary ZDDs to 3-way ZDDs is described here. Given a ZDD, we traverse all paths from the root to $\top$ and by creating a linked list for each such path, we obtain the corresponding list of lists. We then construct a 3-way ZDD from the list of lists. This construction can be done in a similar way to the ZDD construction method based on a sort algorithm [14]. A naive method to convert 3-way ZDDs to ordinary ZDDs is given in the same way.

As shown in Fig. 15, for all datasets except for the random datasets, our conversion algorithms are more efficient than the naive methods, although the difference is not large, particularly for T2Z. It would be challenging to further improve our algorithms. The reason for poor performance in the random

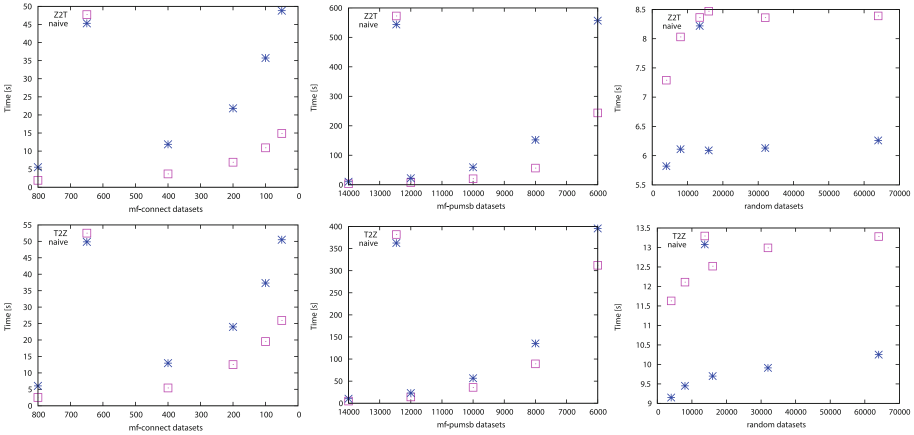


Fig. 15. Comparison of running time for conversion algorithms between 3-way ZDDs and ordinary ZDDs.

datasets is that the random datasets show almost no effect from compression (see Fig. 12). This, perhaps, leads to poorer performance because our algorithm exploits compression efficiency, as can be observed in Fig. 14.

6 Conclusion

We proposed a new data structure, the 3-way ZDD. This data structure can be considered as a variant of ZDDs. We presented conversion algorithms between 3-way ZDDs and ordinary ZDDs, and we analyzed their time-complexity. We conducted experiments using large-scale sparse datasets, and we observed the following things. In many datasets, 3-way ZDDs were much smaller size than a naive representation (i.e., a list of lists) and ternary search trees, and had almost the same size as ordinary ZDDs. The maximum lengths of paths in 3-way ZDDs were reduced and the degradation of performance for membership query was prevented. Our conversion algorithms were faster than naive methods, although the difference was small.

Future work will focus on improving our conversion algorithms, comparing our method to Z-skip-links [10], and applying our method to various problems in data mining.

References

1. Goethals, B.: Survey on frequent pattern mining. Technical report (2002)
2. Minato, S.: Zero-suppressed BDDs for set manipulation in combinatorial problems. In: DAC '93: Proceedings of the 30th International Design Automation Conference, pp. 272–277. ACM, New York (1993)

3. Minato, S., Uno, T., Arimura, H.: LCM over ZBDDs: fast generation of very large-scale frequent itemsets using a compact graph-based representation. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 234–246. Springer, Heidelberg (2008)
4. Minato, S.: A fast algorithm for cofactor implication checking and its application for knowledge discovery. In: Proceedings of IEEE 8th International Conference on Computer and Information Technology, July 2008, pp. 53–58 (2008)
5. Minato, S.: Finding simple disjoint decompositions in frequent itemset data using zero-suppressed BDDs. In: Proceedings of IEEE ICDM 2005 Workshop on Computational Intelligence in Data Mining, November 2005, pp. 3–11 (2005)
6. Minato, S., Ito, K.: Symmetric item set mining method using zero-suppressed BDDs and application to biological data. Trans. Jpn. Soc. Artif. Intell. **22**(2), 156–164 (2007)
7. Coudert, O.: Solving graph optimization problems with ZBDDs. In: Proceedings of the 1997 European Conference on Design and Test, March 1997, pp. 224–228 (1997)
8. Loekito, E., Bailey, J.: Fast mining of high dimensional expressive contrast patterns using zero-suppressed binary decision diagrams. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 307–316. ACM (2006)
9. Minato, S.: Techniques of BDD/ZDD: brief history and recent activity. IEICE Trans. Inf. Syst. **E96-D**(7), 1419–1429 (2013)
10. Minato, S.-I.: Z-Skip-Links for fast traversal of ZDDs representing large-scale sparse datasets. In: Bodlaender, H.L., Italiano, G.F. (eds.) ESA 2013. LNCS, vol. 8125, pp. 731–742. Springer, Heidelberg (2013)
11. Bentley, J.L., Sedgewick, R.: Fast algorithms for sorting and searching strings. In: Proceedings of 8th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '97), January 1997, pp. 360–369 (1997)
12. Knuth, D.E.: The Art of Computer Programming, vol. 4a. Addison-Wesley Professional, New Jersey (2011)
13. Bentley, J.L., Saxe, J.B.: Algorithms on vector sets. SIGACT News **11**(2), 36–39 (1979)
14. Toda, T.: Fast compression of large-scale hypergraphs for solving combinatorial problems. In: Fürnkranz, J., Hüllermeier, E., Higuchi, T. (eds.) DS 2013. LNCS, vol. 8140, pp. 281–293. Springer, Heidelberg (2013)

Local Feature Selection by Formal Concept Analysis for Multi-class Classification

Madori Ikeda(✉) and Akihiro Yamamoto

Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
m.ikeda@iip.ist.i.kyoto-u.ac.jp, akihiro@i.kyoto-u.ac.jp

Abstract. In this paper, we propose a multi-class classification algorithm to apply it to data sets increasing frequently. The algorithm performs lazy learning based on formal concept analysis. We designed it so that it obtains localness in predicting classes of test data and feature selection simultaneously. From a given data set that consists of a set of training data and a set of test data, the algorithm generates a single formal concept lattice. Every formal concept in the lattice represents a cluster of data that are generated by various feature selections. In order to classify each test datum, plausible clusters are selected and combined into a set of neighbors for the test datum. Our algorithm can construct sets of neighbors for test data that are never generated by other algorithms, e.g., the k-nearest neighbor algorithm and decision tree classifiers. We compare our algorithm with other algorithms by experiments using UCI datasets and show that ours is comparable to the others at the viewpoint of correctness.

Keywords: Lazy learning · Multi-class classification · Formal concept analysis · Feature selection

1 Introduction

Multi-class classification is one of the fundamental problems in machine learning. In applying it to real problems, for example, character recognition and disease diagnosis, it should be taken into account that sets of data became larger by taking in newer ones after another. In this paper, we propose a multi-class classification algorithm in order to apply it to such frequently increasing data. Our key idea is to give our algorithm some localness in predicting classes of test data and feature selection simultaneously by adopting *formal concept analysis* (FCA for short) [4,5].

In order to keep the efficiency of classifying such growing data sets, it is important to treat only new data added to the original sets, not the whole of the data. *Lazy learning* (or *instance based learning*) is a popular concept based on this idea. In the context of lazy learning, classification is not executed until at least one new test datum to be classified is given, and only the class of the datum is predicted by referring a few training data. Moreover, no classification

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 470–482, 2014.
DOI: 10.1007/978-3-319-13186-3_42

rules are achieved by using all training data. Lazy learning methods thus can deal with update of training data successfully by replacing simply old ones with new ones. This property is sometimes called *local approximation* of lazy learning.

Feature selection [7], which is data preprocessing for making learning faster and improving learning results, means to select features that correspond to classes of training data by using the whole of the training data. It would be clear that the selection is useful in lazy learning. However, the local approximation, which is the advantage of lazy leaning, focuses only on data, not on features representing the data. Because lazy learning uses a few training data for classifying each test datum, it must improve the classification results to decide a feature space, which is related to decision of the few training data, for each test datum.

We solve this problem by adopting FCA to our novel algorithm called *Concept Lattice-based Classification algorithm* (CLC for short). CLC performs lazy learning and locally selects features for a given test datum in a classification process. In the classification process, it constructs a single formal concept lattice from training data and test data, and formal concepts in the lattice represent clusters of data based on various feature selections. In other words, every formal concept can be regarded as a pair of a set of selected features and a set of data that are similar to each other in the feature space. Finally, test data are classified by evaluating the plausibleness of each formal concept.

This paper is organized as follows. In the next section, we first formalize a multi-class classification problem and then introduce FCA. In Sect. 3, we explain CLC. In Sect. 4, we describe differences among CLC and related works. We compare correctness of classification results of CLC with the results of others by experiments in Sect. 5. Conclusions are placed in Sect. 6.

2 Multi-class Classification and FCA

2.1 Multi-class Classification Problem

We formalize a *multi-class classification problem* and give an example of the problem that is used in the followings.

Let F be a finite set of *features*, and let $n = |F|$. Suppose that each feature $f \in F$ defines a *domain* D_f. Every *datum* x is an element of $\prod_{i=1}^{n} D_{f_i}$ and represented as $(f_1(x), f_2(x), ..., f_n(x))$ where $f_i(x) \in D_{f_i}$ indicates the value of the feature $f_i \in F$ of x. Let L be a finite set of *labels* satisfying $L \cap F = \emptyset$. Each label indicates a *class*. A *training data set* is a triplet $(X, L, \mathcal{L})$ where X is a set of data, $\mathcal{L} : X \rightarrow L$. Every element of X is called a *training datum*. A set Y of data is assumed to be disjoint of X, and every element of Y is called a *test datum*. We also assume that a *target classification rule* $\mathcal{L}_* : \prod_{i=1}^{n} D_{f_i} \rightarrow L$, and that $\forall x \in X. \mathcal{L}_*(x) = \mathcal{L}(x)$. The label $\mathcal{L}_*(x) \in L$ indicates the *true class* of a datum x.

Definition 1. A multi-class classification problem is obtaining a function $\hat{\mathcal{L}} : X \cup Y \rightarrow L$ called a *classifier* from a given training data set $(X, L, \mathcal{L})$ and a given set Y of test data. The classifier is *correct* if $\forall x \in X \cup Y. \hat{\mathcal{L}}(x) = \mathcal{L}_*(x)$.

Table 1. A training data set $(X, L, \mathcal{L})$

$x \in X$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$\mathcal{L}(x)$
x_1	v_{11}	v_{22}	v_{31}	v_{42}	l_1
x_2	v_{11}	v_{22}	v_{31}	v_{42}	l_1
x_3	v_{11}	v_{22}	v_{31}	v_{42}	l_2
x_4	v_{12}	v_{22}	v_{31}	v_{41}	l_2
x_5	v_{12}	v_{22}	v_{31}	v_{41}	l_3
x_6	v_{12}	v_{22}	v_{31}	v_{41}	l_3
x_7	v_{12}	v_{22}	v_{31}	v_{41}	l_4
x_8	v_{12}	v_{21}	v_{32}	v_{43}	l_1
x_9	v_{12}	v_{21}	v_{32}	v_{43}	l_2
x_{10}	v_{12}	v_{21}	v_{32}	v_{43}	l_2

Table 2. A test datum $y \in Y$

$y \in Y$	$f_1(y)$	$f_2(y)$	$f_3(y)$	$f_4(y)$
y	v_{11}	v_{21}	v_{31}	v_{41}

Example 1. As a running example, we give a training data set $(X, L, \mathcal{L})$ where $X = \{x_1, x_2, ..., x_{10}\}$ and $L = \{l_1, l_2, l_3, l_4\}$, and give a set of test data $Y = \{y\}$. Tables 1 and 2 respectively show the training data set and the test datum. In the definition above, the domain of each feature can be any set, nominal (categorical) or numerical, but in this paper we treat only features having a nominal domain. In this example, we let $F = \{f_1, f_2, f_3, f_4\}$, $D_{f_1} = \{v_{11}, v_{12}\}$, $D_{f_2} = \{v_{21}, v_{22}\}$, $D_{f_3} = \{v_{31}, v_{32}\}$, and $D_{f_4} = \{v_{41}, v_{42}, v_{43}\}$. The goal of this classification problem is obtaining a classifier $\hat{\mathcal{L}}$ and predicting the true class of y as $\hat{\mathcal{L}}(y)$.

2.2 Formal Concepts and Formal Concept Lattices

In our algorithm proposed in the next section, a given training data set and a given set of test data are treated as a single data set called a *formal context*. From the formal context, a *formal concept lattice*, which is an ordered set of *formal concepts*, is constructed for classifying every test datum in the context. We introduce the definitions of formal concepts and formal concept lattices with referring to [4,5].

Definition 2 [5]**.** Let G and M be mutually disjoint finite sets, and $I \subseteq G \times M$. Each element of G is called an *object*, and each element of M is called an *attribute*. With a formula $(g, m) \in I$, we intend the g has the attribute m. A triplet (G, M, I) is called a *formal context*.

For a set of nominal data $N \subseteq \prod_{i=1}^{n} D_{f_i}$, there is a simple way to translate N into a context (G, M, I): the set G of objects is the set N of data, the set M of attributes is $\bigcup_{i=1}^{n} D_{f_i}$, and $I = \{(x, m) \in G \times M \mid \exists f \in F. f(x) = m\}$. A set of numerical data also can be translated into a context by a method proposed

Table 3. A formal context $K = (G, M, I)$

	v_{11}	v_{12}	v_{21}	v_{22}	v_{31}	v_{32}	v_{41}	v_{42}	v_{43}
	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9
x_1	×			×	×			×	
x_2	×			×	×			×	
x_3	×			×	×			×	
x_4		×		×	×		×		
x_5		×		×	×		×		
x_6		×		×	×		×		
x_7		×		×	×		×		
x_8		×	×			×			×
x_9		×	×			×			×
x_{10}		×	×			×			×
y	×		×		×		×		

in [6]. In this paper, we regard every datum as an object and every feature as an attribute.

Example 2. Table 3 shows a context $K = (G, M, I)$ which is transformed from a union $G = X \cup Y$ of the nominal data sets X and Y given in Example 1. In the table, for the convenience of the discussion later, every attribute is renamed as m_i, and every element of I is represented with a cross. For example, the datum x_1 has the features $m_1 = v_{11}$, $m_4 = v_{22}$, $m_5 = v_{31}$, and $m_8 = v_{42}$.

Definition 3 [5]**.** For a subset of objects $A \subseteq G$ and a subset of attributes $B \subseteq M$ of a formal context (G, M, I), we define $A^I = \{m \in M \mid \forall g \in A. (g, m) \in I\}$, $B^I = \{g \in G \mid \forall m \in B. (g, m) \in I\}$. A *formal concept* of the context is a pair (A, B) such that $A^I = B$ and $A = B^I$. For every object $g \in G$ of (G, M, I), the formal concept $(\{g\}^{II}, \{g\}^I)$ is called the *object concept* and denoted by γg.

Definition 4 [5]**.** For a formal concept $c = (A, B)$, A and B are called the *extent* and the *intent*, respectively, and let $\mathrm{Ex}(c) = A$ and $\mathrm{In}(c) = B$. For arbitrary formal concepts c and c', we define an order $c \leq c'$ iff $\mathrm{Ex}(c) \subseteq \mathrm{Ex}(c')$ (or equally $\mathrm{In}(c) \supseteq \mathrm{In}(c')$). It is clear that $\leq$ is a partial order. The set of all formal concepts of a context $K = (G, M, I)$ with the order $\leq$ is denoted by $\underline{\mathfrak{B}}(G, M, I)$ (for short, $\underline{\mathfrak{B}}(K)$) and is called the *formal concept lattice* (concept lattice for short) of K.

Definition 5 [5]**.** For every formal concept $c \in \underline{\mathfrak{B}}(K)$, the subset of formal concepts $\{c' \in \underline{\mathfrak{B}}(K) \mid c' \geq c\}$ is denoted by $\uparrow c$ and called the *upper set* of c (also known as the *principal filter* in FCA).

Example 3. Figure 1 shows the concept lattice $\underline{\mathfrak{B}}(K)$ of the context $K = (G, M, I)$ given in Table 3. Each circle represents a formal concept $c \in \underline{\mathfrak{B}}(K)$ with

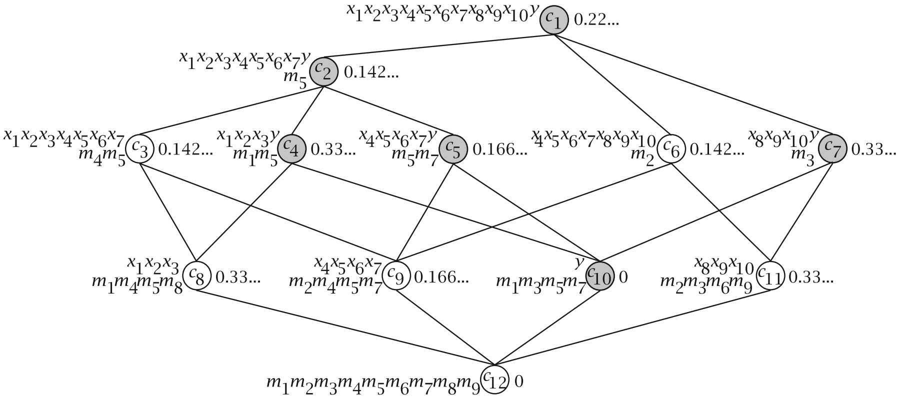


Fig. 1. A concept lattice $\underline{\mathfrak{B}}(K)$

Ex(c) and In(c) on its side. The numbers called *scores* beside the concepts are explained later. In the figure, each edge represents an order $\leq$ between two concepts, and the greater concept is drawn above, and transitional orders are omitted. For this concept lattice $\underline{\mathfrak{B}}(K)$, $\gamma y = c_{10}$ and $\uparrow\gamma y = \{c_1, c_2, c_4, c_5, c_7, c_{10}\}$. Every element in $\uparrow\gamma y$ is shaded in the figure.

Many algorithms have been proposed for enumeration of formal concepts of a context $K = (G, M, I)$, and one of the fastest algorithm [8] can output concepts with $O(\Delta^3)$ delay where $\Delta = \max\{|\{h\}^I| \mid h \in G \cup M\}$. Using the algorithm, constructing a concept lattice takes less than $O(\Delta^3 N_c)$ where $N_c = |\underline{\mathfrak{B}}(K)|$. In addition, several algorithms have been proposed [3,11,13] in order to update the concept lattice $\underline{\mathfrak{B}}(K)$ whenever a new datum or a new feature is added to the context K. For example, K turns into (G', M', I') where $G' \supset G$, $M' \supset M$, and $I' \supset I$. Thus we can easily modify a concept lattice by using these algorithms.

3 A Multi-class Classification Algorithm Using FCA

Our *Concept Lattice-based Classification algorithm* (CLC), which obtains a classifier $\hat{\mathcal{L}}$ for a given training data set $\tau = (X, L, \mathcal{L})$ and a given set of test data Y, is shown in Algorithm 1. The point of CLC is to construct a set of *neighbors* for each test datum based on *local features* in order to classify it. Such sets of neighbors are obtained by combining some clusters of training data that are extents of formal concepts. Thus, the algorithm firstly constructs a formal context $K = (G = X \cup Y, M, I)$ from the given data sets and executes clustering the data in G by constructing $\underline{\mathfrak{B}}(K)$ in the lines 2 and 3.

Every formal concept $c \in \underline{\mathfrak{B}}(K)$ can be regarded as representation of a cluster Ex(c) in which data have features in In(c) in common. In other words, each concept $c = (\text{Ex}(c), \text{In}(c))$ shows a set In(c) of selected features and a cluster Ex(c) generated by filtering G with the selected features. In addition, for every

Algorithm 1. $CLC(\tau, Y)$

```
Require: a training data set $\tau = (X, L, \mathcal{L})$, a set of test data $Y$
Ensure: a classifier $\hat{\mathcal{L}} : X \cup Y \to L$
CLC($\tau, Y$)
    construct a formal context $K = (G = X \cup Y, M, I)$
    construct a concept lattice $\underline{\mathfrak{B}}(K)$
    $\hat{\mathcal{L}}(g) \leftarrow$ Nil for all $g \in G$
    for all $x \in X$ do
        $\hat{\mathcal{L}}(x) \leftarrow \mathcal{L}(x)$
    for all $y \in Y$ do
        $\mathrm{p}(y, \tau) \leftarrow \emptyset$
        for all $c \in \uparrow\gamma y$ do
            get the score $\sigma(c, \tau)$ of $c$
            if $\forall c' \in \mathrm{p}(y, \tau).\, \sigma(c, \tau) > \sigma(c', \tau)$ then
                $\mathrm{p}(y, \tau) \leftarrow \{c\}$
            else if $\forall c' \in \mathrm{p}(y, \tau).\, \sigma(c, \tau) = \sigma(c', \tau)$ then
                $\mathrm{p}(y, \tau) \leftarrow \mathrm{p}(y, \tau) \cup \{c\}$
        $\mathrm{N}(y, \tau) \leftarrow \emptyset$
        for all $c \in \mathrm{p}(y, \tau)$ do
            $\mathrm{N}(y, \tau) \leftarrow \mathrm{N}(y, \tau) \cup \mathrm{Ex}(c, \tau)$
        $\hat{\mathcal{L}}(y) \leftarrow$ the most common class in $\mathrm{N}(y, \tau)$
    return $\hat{\mathcal{L}}$
```

test datum $y \in Y$, a set of formal concepts $\uparrow\gamma y$ is a set of pairs of a set of selected features and a cluster generated by the selection, and there is no formal concept c such that $\mathrm{Ex}(c) \ni y$ out of $\uparrow\gamma y$. Thus, $\uparrow\gamma y$ can be regarded as the whole of the meaningful clusters that are generated based on features selected locally for the test datum y. We therefore think the construction of a concept lattice to be clustering based on *local feature selection*. In the followings, we do not distinguish a formal concept and a cluster represented by the concept.

Example 4. In Fig. 1, the formal concept c_5 in the lattice $\underline{\mathfrak{B}}(K)$ represents a cluster $\{x_4, x_5, x_6, x_7, y\}$ as its extent, and the cluster generated by its intents: every datum in the extent has features m_5 and m_7 in common.

In order to classify each test datum, CLC constructs a set of neighbors by combining *plausible clusters* that are selected from the meaningful clusters based on local features. For representing the plausibleness of each cluster, we define *scores* for every formal concepts.

Definition 6. For every cluster $c \in \underline{\mathfrak{B}}(K)$ and a training data set $\tau = (X, L, \mathcal{L})$, we define $\mathrm{Ex}(c, \tau) = \mathrm{Ex}(c) \backslash X$. We define a real number $\sigma(c, \tau)$ in $[0, 1]$, called the *score* of the cluster c under the training data set τ, as follows:

$$\sigma(c, \tau) = \begin{cases} 0 & \text{if } |\mathrm{Ex}(c, \tau)| \ = 0, \\ 1 & \text{if } |\mathrm{Ex}(c, \tau)| \ = 1, \text{and} \\ \dfrac{|\{(x_i, x_j) \in \mathrm{Ex}(c, \tau)^2 \mid i < j, \mathcal{L}(x_i) = \mathcal{L}(x_j)\}|}{|\{(x_i, x_j) \in \mathrm{Ex}(c, \tau)^2 \mid i < j\}|} & \text{otherwise.} \end{cases}$$

Because a cluster should be more plausible when it is useful to classify a test datum correctly, the score is calculated with the training data set. The function σ calculates the average of similarities among training data in $\mathrm{Ex}(c, \tau)$ and is designed to estimate similarity among all data in $\mathrm{Ex}(c)$, which includes not only training data but also test data. Then, plausible clusters are selected based on the scores.

Definition 7. For every test datum $y \in G$ in a concept lattice $\underline{\mathfrak{B}}(G, M, I)$ and a training data set $\tau = (X, L, \mathcal{L})$, a cluster $c \in \uparrow\gamma y$ is called *plausible* w.r.t. y and τ if $\sigma(c, \tau) \geq \sigma(c', \tau)$ for any other cluster $c' \in \uparrow\gamma y$. The set of plausible clusters w.r.t. y and τ is denoted by $\mathrm{p}(y, \tau) \subseteq \uparrow\gamma y$.

The decision of each plausible cluster $p \in \mathrm{p}(y, \tau)$ can be regarded as selecting one of the best sets of features $\mathrm{In}(p)$ filtering data so that the filtered data in $\mathrm{Ex}(p)$ can be estimated to be similar to each other under the given training data set τ.

For classifying each test datum $y \in Y$, the algorithm selects plausible clusters $\mathrm{p}(y, \tau)$ in the lines 8–14, and it constructs a set of neighbors $\mathrm{N}(y, \tau)$ by combining the selected plausible clusters in the lines 15–17. We claim that the set of neighbors $\mathrm{N}(y, \tau)$ of y is a cluster generated by using multiple feature selection because plausible clusters composing the neighbors are generated from different sets of local features. Finally, at the line 18, the class $\hat{\mathcal{L}}(y)$ of each test datum $y \in Y$ is predicted. Note that, if some classes are the most frequent among training data in $\mathrm{N}(y, \tau)$, one of them is randomly selected for y.

Example 5. In Fig. 1, each number next to each formal concept represents its score under the classes of training data given in Table 1. According to the scores of clusters c_4 and c_5, a set of features $\mathrm{In}(c_4)$ is more plausible than a set of $\mathrm{In}(c_5)$ for the sake of generating a cluster in which data are similar to each other. The scores indicate that the clusters c_4 and c_7 are plausible for the test datum y under the given training data set, and a set of neighbors $\mathrm{Ex}(c_4, (X, L, \mathcal{L})) \cup \mathrm{Ex}(c_7, (X, L, \mathcal{L})) = \{x_1, x_2, x_3, x_8, x_9, x_{10}\}$ are constructed for y. As shown in Table 1, classes l_1 and l_2 are the most common among the neighbors, so one of them are randomly picked up as a predicted class $\hat{\mathcal{L}}(y)$.

The time complexity of CLC for classifying all data in $G = X \cup Y$ with a given training data set $(X, L, \mathcal{L})$ is $O(\varDelta^3 N_c) + O(N_c N_x^2)$ where $N_x = |X|$. The first term is the time complexity of constructing a concept lattice in the lines 2 and 3. The time complexity of constructing $\hat{\mathcal{L}}(x)$ for every training datum $x \in X$ is $O(N_x)$. For test data, scoring a cluster takes less than $O(N_x^2)$, and all of N_c clusters need to be scored in the worst case. Constructing a set of neighbors and predicting a class take less than $O(N_c)$ and $O(N_x)$ respectively for each test datum $y \in Y$. Thus, the process for constructing $\hat{\mathcal{L}}(g)$ for every datum $g \in G$ takes less than $O(N_x) + O(N_c N_x^2) + O(|Y|N_c) + O(|Y|N_x) = O(N_c N_x^2)$.

While CLC constructs no classification rules, it can update the classifier whenever original data sets are updated. This means that the algorithm classifies a new test datum immediately, and that it revises classes of test data that are already predicted when a given training data set is changed. After the classifier

$\hat{\mathcal{L}}$ is obtained, CLC keeps the lattice $\underline{\mathfrak{B}}(K)$. When the context K is changed into a new context K' by update, the algorithm changes $\underline{\mathfrak{B}}(K)$ into $\underline{\mathfrak{B}}(K')$ by using proposed methods [3,11,13]. If new test data are added by the update, CLC executes the lines 8–18 for each of the new test data by using $\underline{\mathfrak{B}}(K')$. In the case that existing training data are changed or new training data are added, CLC calculates scores for every clusters $c \in \uparrow\gamma x'$ for each datum x' of such training data in $\underline{\mathfrak{B}}(K')$. Then, it executes the lines 8–18 for every test datum $y \in \mathrm{Ex}(c')$ where $c' \in \underline{\mathfrak{B}}(K')$ is a newly scored cluster.

4 Related Works

The *k-nearest neighbors algorithm* (k-NN for short) [2] is similar to ours, CLC, at the point of deciding neighbors for each test datum in order to classify it. They are used in *lazy learning* in which any classifier $\hat{\mathcal{L}}$ is not constructed until test data to be classified are given, and then a class $\hat{\mathcal{L}}(y)$ is predicted for each test datum y. Thus, the target function $\mathcal{L}_*$ will be approximated locally by focusing on each test datum y. Neighbors affecting the approximation of the target are also decided for each test datum based on the distance on the feature space. However, the feature space is not selected for each test datum in k-NN. In CLC, various feature spaces are generated for each test datum, and each of the spaces are represented as the intent of each formal concept. Consequently, CLC can generate even a set of neighbors which is never made by k-NN for a test datum.

Example 6. Using the symmetric difference, a distance $\delta(g, g')$ between two objects g and g' of a context (G, M, I) can be defined as $\delta(g, g') = |\{g\}^I \cup \{g'\}^I| - |\{g\}^I \cap \{g'\}^I|$. According to Table 3, training data x_1, x_2, ..., x_7 are placed so that the distance between each of them and the test datum y is 4. While these training data can not be distinguished by k-NN, CLC distinguishes these and separates them into two clusters $\mathrm{Ex}(c_4)$ and $\mathrm{Ex}(c_5)$ that can be a set of neighbors of y alone. Moreover, even a set of neighbors $\{x_1, x_2, x_3, x_8, x_9, x_{10}\}$ can be constructed in which, as given in Example 2, composing elements are more distant from y than outer elements: the distance between y and every of x_8, x_9, x_{10} is 6, and the distance between y and every of x_4, x_5, x_6, x_7 is 4.

CLC is also similar to *decision tree methods* [10] because we can easily transform a concept lattice into a tree. In the decision tree method, a feature more suitable for representing classes of training data is prior to others and is used in constructing a decision tree, and features are used as decision rules to classify test data. Training data classified into the same leaf that contains a test datum by decision rules are used like neighbors in the nearest neighbors algorithm. In other words, training data are separated by decision rules in order to generate sets of neighbors. In CLC, when intents of formal concepts are regarded as decision rules, each upper set $\uparrow\gamma y$ is a decision tree for the test datum y. However, the decision tree method only separates training data. Some of sets of separated training data called clusters are sometimes combined into a set of neighbors in CLC. Because of the combination, a classification result by CLC differs from one by the decision tree method.

Example 7. Figure 2 shows a decision tree constructed based on *information gain* [10] from the training data set in Table 3. Using this tree, the example test datum y is decided to be similar to training data x_4, x_5, x_6, x_7 and classified into the class l_3. This result is different from one of ours given in Example 5.

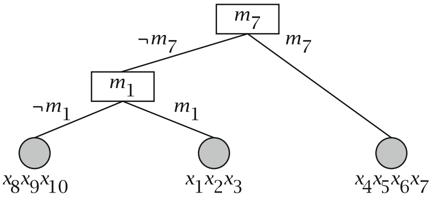


Fig. 2. A decision tree constructed from the example training data set

5 Experiments

We evaluate our algorithm by comparing with other algorithms in experiments.

5.1 Data Sets and Algorithms

In the experiments, six nominal data sets provided from UCL Machine Learning Repository [1] are used, and all of them are translated into formal contexts by the way described in Sect. 2.2. In the translation of the data sets, missing values are treated as the same value. Statistics for the translated data sets are shown in Table 4.

Table 4. Statistics for UCI datasets used in our experiments

Data set	#Data	#Classes	#Features
Balance Scale	625	3	20
Car Evaluation	1,728	4	21
Congressional Voting Records (CVR)	435	2	48
Hayes Roth	132	3	15
Mushroom	8,124	2	117
Nursery	12,960	5	27

Six multi-class classification algorithms are executed: our algorithm (CLC), the k-nearest neighbor algorithm for $k = 1, 10$ (1-NN, 10-NN), the naïve bayes classifier (NB), the support vector machine (SVM), the decision tree method using information gain (Tree). In the first step of CLC, we used LCM[1] [12] in order to enumerate all formal concepts, which was implemented in C. Both of the rest of CLC and the other algorithms are implemented in Python version 2.7.5. Excepting CLC, all algorithms are provided by Scikit-learn version 0.14 [9], which is a machine learning library.

[1] http://research.nii.ac.jp/~uno/codes.htm

5.2 Results

Figure 3 shows correctnesses of experimental results of the algorithms. Each correctness data point is calculated as the average of ratios of test data that are classified correctly by using 10-fold cross validation. CLC performs better than each of 1-NN and 10-NN in all data set. CLC works correctly than NB except for the results in *Balance Scale* data set, and performances of them are almost equivalent in the data set. Compered with SVM, CLC is better in *Car Evaluation* and *Nursery*, but worse in the others. This might be related with that SVM is originally designed for binary classification because the numbers of classes in

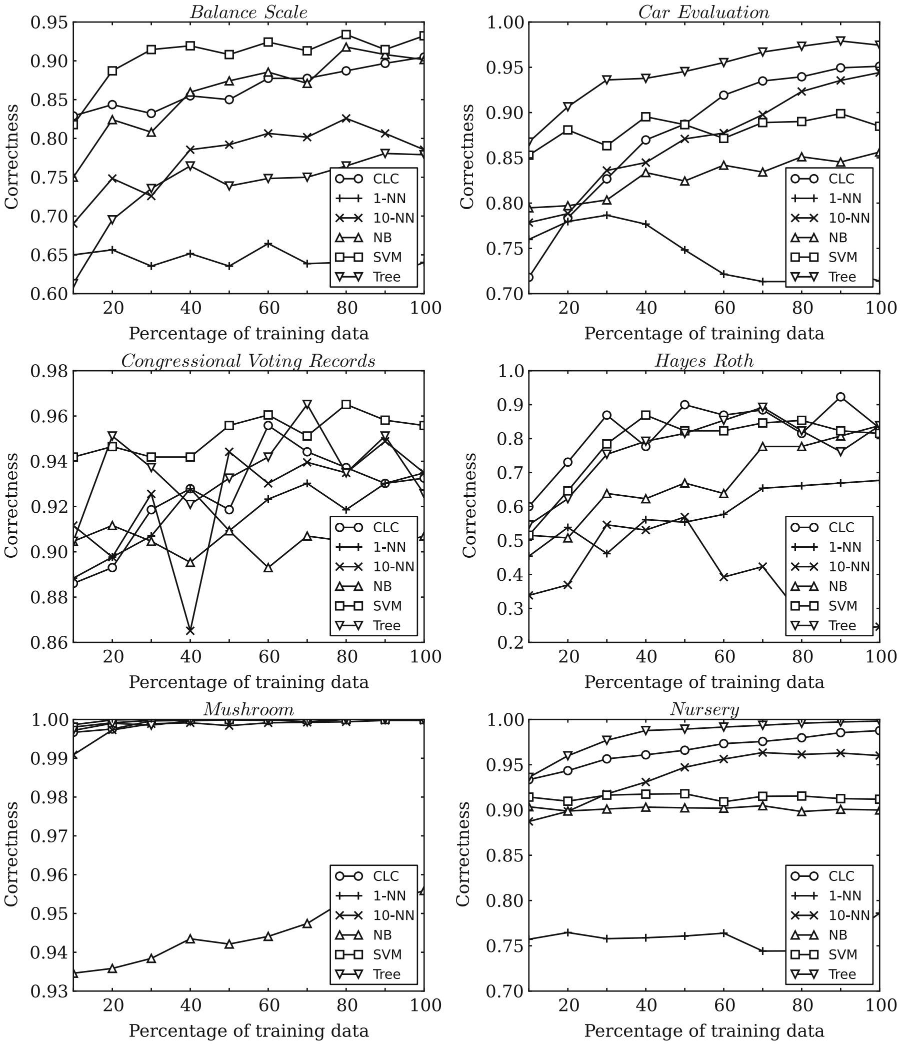


Fig. 3. Correctnesses of experimental results

Table 5. Comparison of the numbers of neighbors for each test datum

Data set	Algorithm	#Test data	#Same	#Different	#Larger	#Smaller
Balance Scale	1-NN		0.0	0.0	0.0	62.0
	10-NN	62	0.0	0.0	43.2	18.8
	Tree		0.0	2.4	36.7	22.9
Car Evaluation	1-NN		0.0	0.0	0.0	172.0
	10-NN	172	0.0	0.6	34.4	137.0
	Tree		0.0	1.9	35.2	134.9
CVR	1-NN		0.0	0.0	0.0	43.0
	10-NN	43	0.0	0.0	0.0	43.0
	Tree		0.0	0.1	1.0	41.9
Hayes Roth	1-NN		0.0	0.0	0.0	13.0
	10-NN	13	0.0	3.1	3.0	6.9
	Tree		0.0	1.1	1.4	10.5
Mushroom	1-NN		0.0	0.0	0.0	812.0
	10-NN	812	0.0	0.0	0.0	812.0
	Tree		0.0	0.0	68.1	743.9
Nursery	1-NN		0.0	0.0	0.0	1296.0
	10-NN	1296	0.0	4.9	23.8	1267.3
	Tree		0.0	3.7	416.4	875.9

Car Evaluation and *Nursery* are larger than ones of the others. It is also seen that CLC surpasses Tree in several data sets. Thus, we can conclude that our algorithm, CLC, works as correctly as the other algorithms do in practical use. In particular, CLC is preferable for multi-class classification.

In addition, we compare sets of neighbors of CLC with ones of the other algorithms, 1-NN, 10-NN, and Tree, by experiments. In Sect. 4, we claim that our algorithm constructs sets of neighbors that are never constructed by the others, so it can obtain a classifier which is different from theirs. Table 5 shows the comparison. For the sake of convenience, we express a set of neighbors for a test datum $y \in Y$ in an algorithm a as $N(y, a)$. In each line of the table, from the left to the right, the name of a data set, an algorithm a, and the number of test data $|Y|$ are shown, and then

$$
\begin{aligned}
&|\{N(y,a) \mid N(y,a) = N(y,\mathrm{CLC})\}|,\\
&|\{N(y,a) \mid N(y,a) \neq N(y,\mathrm{CLC}), |N(y,a)| = |N(y,\mathrm{CLC})|\}|,\\
&|\{N(y,a) \mid |N(y,a)| > |N(y,\mathrm{CLC})|\}|, and\\
&|\{N(y,a) \mid |N(y,a)| < |N(y,\mathrm{CLC})|\}|
\end{aligned}
$$

are following. In the experiments adopting 10-fold cross validation, every algorithm uses all training data, and each number in the table is calculated as the average of values of the formulae above. The table proves that neighbors found by CLC are quite different from ones of the other algorithms, and it is clear that the differences directly cause differences among correctnesses of the algorithms.

6 Conclusions

We have proposed a novel algorithm, CLC, for multi-class classification. The algorithm performing lazy learning uses FCA in order to realize not only local approximation but also local feature selection. For every given test datum, it selects features in various ways, it generates various sets of features, and a set of neighbors is found based on the selections by regarding formal concepts as pairs of a subset of features and a set of neighbors. These processes enable the algorithm to generate the set of neighbors which is never found by the k-nearest neighbor algorithm and the decision tree method. By experiments, we can conclude that our algorithm is never inferior to other algorithms from the viewpoint of the correctness of results.

In our future works, we have to improve CLC at the point of complexity. Even though all lazy learning algorithms are inferior to eager learning methods in the time complexity, we must overcome the weakness. In addition, the algorithm needs a memory storage which is large enough to maintain the whole concept lattice. We conjecture that an ad hoc way can solve both of the problems. Time and storage can be reduced at the same time by enumerating only formal concepts needed for construction of a set of neighbors of each test datum.

References

1. Bache, K., Lichman, M.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2013). http://archive.ics.uci.edu/ml
2. Dasarathy, B.V.: Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos (1991)
3. Choi, V., Huang, Y.: Faster algorithms for constructing a galois lattice, enumerating all maximal bipartite cliques and closed frequent sets. In: SIAM Conference on Discrete Mathematics (2006)
4. Davey, B.A., Priestly, H.A.: Introduction to Lattice and Order. Cambridge University Press, Cambridge (2002)
5. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York Inc., Secaucus (1999)
6. Kaytoue, M., Kuznetsov, S.O., Napoli, A., Duplessis, S.: Mining gene expression data with pattern structures in formal concept analysis. J. Inf. Sci. **181**(10), 1989–2001 (2011)
7. Kira, K., Rendell, L.A.: The feature selection problem: traditional methods and a new algorithm. In: Proceedings of AAAI'92, pp. 124–134 (1992)
8. Makino, K., Uno, T.: New algorithms for enumerating all maximal cliques. In: Hagerup, T., Katajainen, J. (eds.) SWAT 2004. LNCS, vol. 3111, pp. 260–272. Springer, Heidelberg (2004)

9. Pedregosa, F., et al.: Scikit-learn: machine learning in python. JMLR **12**, 2825–2830 (2011)
10. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
11. Soldano, H., Ventos, V., Champesme, M., Forge, D.: Incremental construction of alpha lattices and association rules. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) KES 2010, Part II. LNCS, vol. 6277, pp. 351–360. Springer, Heidelberg (2010)
12. Uno, T., Kiyomi, M., Arimura, H.: LCM ver. 3: collaboration of array, bitmap and prefix tree for frequent itemset mining. In: Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations, pp. 77–86. ACM (2005)
13. Valtchev, P., Missaoui, R.: Building concept (Galois) lattices from parts: generalizing the incremental methods. In: Proceedings of the ICCS'01, pp. 290–303 (2001)

Ensemble Clustering of High Dimensional Data with FastMap Projection

Imran Khan[1](✉), Joshua Zhexue Huang[1,2], Nguyen Thanh Tung[1], and Graham Williams[1]

[1] Shenzhen Key Laboratory of High Performance Data Mining,
Shenzhen Institutes of Advanced Technology,
Chinese Academy of Sciences, Shenzhen 518055, China
imran.khan@siat.ac.cn, zx.huang@szu.edu.cn,
tungnt@wru.vn, Graham.Williams@togaware.com

[2] College of Computer Science and Software Engineering,
Shenzhen University, Shenzhen 518060, China

Abstract. In this paper, we propose an ensemble clustering method for high dimensional data which uses FastMap projection to generate subspace component data sets. In comparison with popular random sampling and random projection, FastMap projection preserves the clustering structure of the original data in the component data sets so that the performance of ensemble clustering is improved significantly. We present two methods to measure preservation of clustering structure of generated component data sets. The comparison results have shown that FastMap preserved the clustering structure better than random sampling and random projection. Experiments on three real data sets were conducted with three data generation methods and three consensus functions. The results have shown that the ensemble clustering with FastMap projection outperformed the ensemble clusterings with random sampling and random projection.

Keywords: Ensemble clustering · FastMap · Random sampling · Random projection · Consensus function

1 Introduction

The emergence of new application domains results in very high dimensional big data such as text data, microarray data and smart phone user behavior data. Such high dimensional data with thousands of features present a big challenge to current data mining techniques [10]. Curse of dimensionality and sparsity are two main problems, among others, that handicap many clustering algorithms to find strongly cohesive clusters from very high dimensional big data.

Ensemble clustering [14,15] is a new approach for clustering that integrates results of different clusterings from the same original data generated by different clustering algorithms or from different data sets sampled from the original data. The ensemble clustering can have a higher accuracy than individual component

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 483–493, 2014.
DOI: 10.1007/978-3-319-13186-3_43

clustering results. A large number of research results have accelerated this field [6]. Different ensemble clustering methods have been proposed to ensemble different types of clustering results like clustering results of one algorithm from the same data with different parameters initialization [13], results from the same data by using different algorithms [1], results of multiple component data sets from the same data set [8]. Subspace ensemble clustering has become a useful strategy to find robust clusters from such sparse and high dimensional data.

In high dimensional data, clusters often exist in different subspaces. Ensemble clustering based on full space clustering algorithms fail to cluster such data. The innovation of subspace ensemble clustering techniques is promised to resolve this problem. Recently, two methods for generating low dimensional component data have been used to resolve the problem of subspace ensemble clustering of high dimensional data clustering. One method generates low dimensional data by randomly sampling different features. The other method generates low dimensional component data by using a random projection matrix to project the original high dimensional data onto a low dimensional space. We call the former random sampling method and the latter random projection method. In recent days, different flavours of random projection are available [11,16] but for ensemble clustering former random projection has been used [4]. Both, random sampling and random projection benefit ensemble clustering for high dimensional sparse data. However, the drawback of these methods is that they cannot well preserve the clustering structure of the original data in their generated low dimensional component data, which increases discrepancy of clustering structures in component data sets, thus affecting the performance of ensemble clustering for high dimensional data.

In this paper, we present a new low dimensional component data generation method by FastMap [5], an algorithm that is used to generate low dimensional transformation of high dimensional data. Given a distance matrix of N objects, FastMap uses the well known Cosine Law to compute the coordinates of the N objects that are projected to the line of two pivot objects selected from the data set. By removing the distance component from the new generated dimension, a new set of coordinates are computed. This process repeats until k dimensional representation of the N objects is obtained. The advantage of FastMap projection in comparison with random sampling and random projection is that it can better preserve the clustering structure of the original data in its generated component data sets. Thus, the performance of ensemble clustering is improved significantly.

We propose two methods to measure preservation of clustering structure of original data in the generated component data sets. We used three real world data sets to analyze preservations by random sampling, random projection and FastMap projection. The comparison results have shown that FastMap preserved clustering structure better than other two methods. We also used the three data sets to conduct ensemble clustering experiments with three component data generation methods and three consensus functions to ensemble clustering results. k-means algorithm was used to generate component clustering.

The results have shown that the ensemble clustering with FastMap projection outperformed the ensemble clusterings with random sampling and random projection on all three data sets. The overall performance of FastMap was the best among the three methods.

2 Framework for Subspace Ensemble Clustering

Ensemble clustering of a data set X is a process to integrate multiple clustering results produced by one or more clustering algorithms from component data sets sampled from X into a single clustering of X with a result that is usually much better than the results of individual clusterings on X [15]. The subspace ensemble clustering framework consists of the following steps.

- **Step1 :** Generate K different component data sets $\{C_1, C_2, \cdots, C_K\}$ from X using a component generation method.
- **Step2 :** Cluster the K component data sets to produce K component clusterings $\{\pi_1, \pi_2, \cdots, \pi_k\}$ independently using one or more clustering algorithms.
- **Step3 :** Ensemble K component clusterings into a single clustering π using an ensemble method called a consensus function.

Figure 1 shows a generic framework of ensemble clustering.

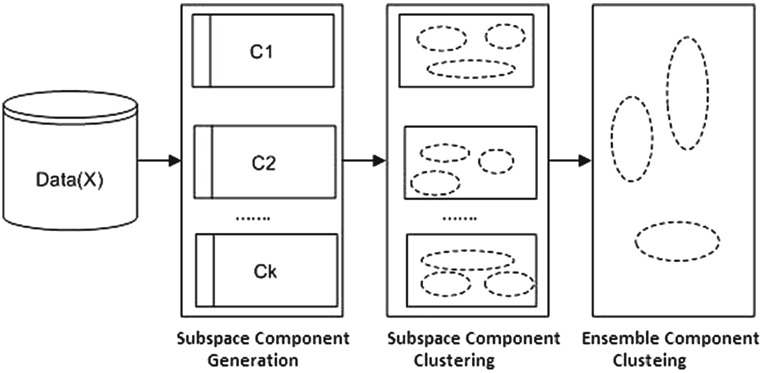

Data(X)
C1
C2
....
Ck
....
Subspace Component Generation
Subspace Component Clustering
Ensemble Component Clusteing

Fig. 1. Generic framework of ensemble component clustering.

2.1 Subspace Component Generation

In ensemble clustering of high dimensional data, we are interested to generating low dimensional component data sets that can better preserve the clustering structure of the original data so as to improve the performance of ensemble clustering on high dimensional data. Currently, random projection and random sampling are two widely used methods for low dimensional component data generation. We review these two method briefly below.

Random Projection. The random projection method projects high dimensional data into low dimensional space by using a random matrix [3]. A data set $X_{N \times m}$ with m dimensions and N objects is projected into a low dimensional subspace data $Y_{N \times p}$ with p≪m, via,

$$Y_{N \times p} = X_{N \times m} \times R_{m \times p} \tag{1}$$

Computationally, random projection cost is O(p × m × N). The value of each element r_{ij} of matrix R should follow Gaussian distribution.

$$r_{ij} = \sqrt{3} \begin{cases} +1 \; with \quad probability \; \frac{1}{6} \\ \;\;0 \; with \quad probability \; \frac{2}{3} \\ -1 \; with \quad probability \; \frac{1}{6} \end{cases} \tag{2}$$

Random Sampling. A number of methods have been proposed to generate low dimensional component data sets by randomly selection of features. These methods differ in the way of selection of dimensions from the original high dimensional data set. A commonly used random sampling is to use a threshold on a feature or set of features, and a feature is selected to add in the component data set if the corresponding value exceeds the threshold value. The features can be selected randomly by using any of the following proposed methods like Gini index, Quinlan's information gain ratio or Mingers's G statistic. In our experiments, we have used Quinlan's information gain ratio for features extraction.

2.2 Component Data Clustering

Any clustering algorithm can be used to cluster a low dimensional component data. Popular clustering algorithms are k-means, subspace k-means and hierarchical clustering methods. The advantage of the k-means type algorithms is its efficiency in handling large data. In this work, we used k-means. Quite often, different clustering algorithms were used to generate different component clustering results for ensemble clustering. However, there is no clear guidance how the different clustering algorithms should be used. In practice, it is more convenient to use one clustering algorithm for ensemble clustering, rather than multiple clustering algorithms.

2.3 Ensemble Component Clusterings

An ensemble method is used to ensemble multiple component clusterings from different component data sets into a single clustering as the final clustering result. In ensemble clustering, ensemble method is also called consensus function. Several consensus functions have been proposed with different strategies and methods to ensemble component clustering results. Below, we briefly review three ensemble methods that were used in this work.

Similarity-Based Consensus Function. A clustering signifies a relationship between objects in the same cluster and can thus be used to establish a measure of pairwise similarity [15]. A similarity matrix for each component clustering is constructed. In the similarity matrix, the element indexed two objects in the same cluster is assigned value 1, otherwise, the element has value 0 if the two objects are in different clusters. After computation of K similarity matrices, a final matrix is obtained as the average of corresponding cells of all similarity matrices. The METIS algorithm [9] is then used to resultant similarity matrix to get final clustering ensemble.

Hyper Graph-Based Consensus Function. In Hyper Graph-based Consensus Function (HGPA), an ensemble problem is formulated as partitioning the hypergraph by cutting a minimal number of hyperedges [15]. The hyper graph is constructed by considering objects of a data set X as N vertices, and hyper-edges with the same weight are used to connect a set of vertices by using K component clusterings. The algorithm HMETIS [12] is used to partition the hyper-graph into unconnected components by cutting a minimum number of hyper-edges.

Meta Cluster-Based Consensus Function. This method was introduced in the meta-clustering algorithm (MCLA) [15]. Similar to HGPA, a hyper graph is constructed by considering clusters of component clusterings as vertices and edges are used to connect these vertices. The weight between two vertices (clusters) is computed by using the following binary Jaccard distance equation.

$$\mathrm{J}acSim(C_x, C_y) = \frac{C_x^{'} C_y}{\|C_x\|_2^2 + \|C_y\|_2^2 - C_x^{\prime} C_y} \tag{3}$$

where $C_x^{'}$ and C_y are two vectors of N elements representing two clusters where N is the number of objects in the data set X. Each element in the vector corresponds to one object. If the cluster contains the object, the corresponding element is assigned to 1. Otherwise, the element is 0. $C_x^{'}$ is the transpose of C_x. METIS is used to partition this hyper graph to identify K meta-clusters. Finally, a voting method is used to assign each data point to its most associated meta-cluster.

3 FastMap Projection for Component Data Generation

3.1 FastMap Projection

FastMap [5] is an efficient algorithm to generate k dimensional coordinates of N objects from a dissimilarity matrix of the N objects. Given a high dimensional data X of m dimensions and N objects, we use a distance function to compute the distance matrix $S_{N \times N}$. We select two objects O_a and O_b with a large distance as pivot objects and take the straight line passing the two objects as the projection axis of the first dimension coordinate F_1. The coordinate of object O_i in the first dimension is computed by using the following cosine equation.

$$\mathrm{x}_i = \frac{d_{a,i}^2 + d_{a,b}^2 - d_{b,i}^2}{d_{a,b}} \tag{4}$$

where $d_{a,b}$ is the distance between the pivot objects O_a and O_b, $d_{a,i}$ is the distance between the pivot object O_a and object O_i and $d_{b,i}$ is the distance between the pivot object O_b and object O_i.

After the coordinates of all N objects are computed, we compute the reduced distance matrix $S^{'}$ of N objects according to *Lemma 1* in [5] as

$$\mathrm{d}^{'}(O_i, O_j)^2 = d(O_i, O_j)^2 - (x_i - x_j)^2 \tag{5}$$

where $d^{'}$ is the reduced distance in $S^{'}_{N\times N}$, d is the distance in $S_{N\times N}$ and the last term on the right is the squared distance in the new dimension.

Given $S^{'}_{N\times N}$, we can choose a new pair of pivot objects and use (4) to compute the coordinates of the second dimension. We repeat this process k times to generate a k dimensional data of X.

We can also use Principal Component Analysis (PCA) to generate a low dimensional data of X. However, given X, we can only use PCA to generate one low dimensional data, thus not suitable for ensemble clustering which requires multiple component data sets. Using FastMap, we can use a random process to select different pairs of pivot objects to produce different projections of data as component data sets. Another advantage of FastMap is that it is efficient to handling large data.

3.2 Evaluation of Component Data Generation

In this section, we present two methods to evaluate component data generation for ensemble clustering, i.e., preservation of clustering structure of the original data in generated component data sets.

Intrinsic Dimensionality. Given a high dimensional data set $X_{N\times m}$ of m dimensions and N objects, we use a component data generation function $\Phi(X, \theta)$ to generate a subspace data $Y_{N\times p}$, i.e., $\Phi(X, \theta) = Y_{N\times p}$ where θ are input parameters to produce different Ys from X. Let $\mathbf{Y} = \{Y_1, \ldots, Y_L\}$ be a set of L component data sets all in p dimensions and $\mathbf{D} = \{D_1, \ldots, D_L\}$ the set of L distance matrices computed from $\mathbf{Y}$. Given a distance matrix D_i, we take the upper half mutual distances of D_i and plot the histogram of the mutual distances. Large mean and small variance of the histogram distribution of D_i represent a problem of *curse of dimensionality*. We use *intrinsic dimensionality* to measure curse of dimensionality of a data set as in [2].

Definition 1. The intrinsic dimensionality of a data set in a metric space is defined as $\rho = \mu^2/2\sigma^2$ where μ and σ are the mean and variance of its histogram of distances.

We use intrinsic dimensionality ρ to evaluate a method $\Phi(X,\theta)$. For each component Y_i, we compute ρ_i. Then, we compute the average $\bar{\rho}$ of ρ_i of L component data sets in $\mathbf{Y}$. The smaller $\bar{\rho}$ the better the method $\Phi(X,\theta)$.

Distance Preservation. Let D be the distance matrix of high dimensional data $X_{N\times m}$ and $\{D_1,\ldots,D_L\}$ the set of L distance matrices computed from $\mathbf{Y}$. Given D and D_i from Y_i, we compute

$$Dist = \sqrt{\frac{\sum_{i=1,j=1}^{N}(d_s(o_i,o_j)-d_o(o_i,o_j))^2}{\sum_{i=1,j=1}^{N} d_o^2(o_i,o_j)}} \tag{6}$$

where N is the total number of objects in the data sets, d_o is the distance between two objects in data set $X_{N\times m}$ and d_s is the distance between corresponding two objects in the subspace component data set Y_i. $Dist$ is a measure of distance preservation of component data set Y generated from X. The smaller $Dist$, the better the component data set Y in preserving the mutual distances of objects in X. Dasgupta's results [7] have shown that random projection preserves the separation among Gaussian clusters having large variances. Random projection can change the shape of highly eccentric clusters to more spherical shape [7].

4 Experiments

In this section, we present experiments on real world data sets to evaluate the performance of ensemble clusterings with FastMap projection. The FastMap results are compared with the results of ensemble clustering with random sampling and random projection. Three high dimensional data sets in different application domains were selected from the available Web sites of UCI machine learning repository and feature selection at Arizona state university. The characteristics of these data sets are listed in Table 1.

Table 1. Real world data sets.

Data sets	#Instances	#Features	Source	#Classes
Internet Ad	1000	1558	Multivariate	02
GLI-85	85	22283	Microarray	02
Orlraws10P	100	10304	Image	10

4.1 Experiment Settings

For each data set, we used three methods, random sampling (RS), random projection (RP) and FastMap projection (FP) to generate component data sets. We used the k-means clustering algorithm to cluster each component data set. The number of clusters k was given as the number of classes in the data set.

For ensemble clustering, we used the three consensus functions discussed in Sect. 3, i.e., hyper graph based consensus function (HGPA), similarity-based consensus function (CSPA) and meta cluster-based consensus function (MCLA). By combining the three component data generation methods and the three consensus functions, we produced 9 ensemble clustering results from each data set. We denote these 9 ensemble clustering methods as RS-CSPA, RP-CSPA, FM-CSPA, RS-MCLA, RP-MCLA, FM-MCLA, RS-HGPA, RP-HGPA and FM-HGPA respectively. We have also shown the ensemble clustering results by using k-means (KM) upon original data.

In conducting the experiments for comparisons, the component data sets from the same data set were generated with the same number of dimensions by each component generation method. Each ensemble clustering was produced from 10 component clusterings which were produced with the k-means algorithm.

4.2 Evaluation Methods

We used four evaluation methods to evaluate the results of ensemble clustering with the 9 ensemble clustering methods. They were one unsupervised method and three supervised methods given below.

The unsupervised evaluation method is cluster compactness (CP) calculated as

$$CP = \frac{1}{n}\sum_{x=1}^{k} n_x \left(\frac{\sum_{o_i,o_j \in C_x} d(o_i,o_j)}{n_x(n_x - 1/2)} \right) \tag{7}$$

where $d(o_i,o_j)$ is the distance between the objects o_i and o_j in a cluster C_x, n_x is the number objects in a cluster C_x. The smaller the value of CP, the better the clustering result.

The three supervised evaluation methods are normalized mutual information (NMI), adjusted rand index (ARI) and clustering accuracy (CA), calculated as follows.

$$\begin{aligned} CA &= \frac{1}{n}\sum_{x=1}^{k} max_y n_{x,y} \\ ARI &= \frac{\sum_{x=1}^{k}\sum_{y=1}^{k}\binom{n_{x,y}}{2} - s_3}{\frac{1}{2}(s_1+s_2) - s_3} \\ NMI &= \frac{\sum_{x=1}^{k}\sum_{y=1}^{k} n_{x,y} log\frac{nn_{x,y}}{n_x n_y}}{\sqrt{\sum_{x=1}^{k} n_x log\frac{n_x}{n}\sum_{y=1}^{k} n_y log\frac{n_y}{n}}} \end{aligned} \tag{8}$$

where n_x and n_y are the total number of objects in cluster x and class y respectively, $n_{x,y}$ is the total number of objects in cluster x and class y, n is the total number of objects in the given data set, $s_1 = \sum_{x=1}^{k}\binom{n_x}{2}$, $s_2 = \sum_{y=1}^{k}\binom{n_y}{2}$ and $s_3 = \frac{2s_1s_2}{n(n-1)}$. The larger the values of these measures, the better the clustering result.

4.3 Experimental Results

Table 2 lists the ensemble clustering results of three data sets produced with 9 ensemble clustering methods. These results were evaluated with 4 evaluation methods. The best results were marked in bold font. We can see that under clustering accuracy evaluation, the ensemble clustering with FastMap projection performed best in all data sets compared with the ensemble clusterings with other two methods with the same consensus function. In all 9 ensemble clustering methods, the highest clustering accuracy was obtained in all data sets by the ensemble clustering with FastMap projection. The improvements in data sets GLI85 and Internet Advertisements were significant compared with other two methods.

Table 2. Clustering result comparison.

Methods	Internet Ad				GLI-85				Orlraws10P			
-	CP	NMI	ARI	CA	CP	NMI	ARI	CA	CP	NMI	ARI	CA
RS-CSPA	171.5	0.01	0.50	0.54	561570	0.45	0.48	0.67	3711	0.67	0.81	0.78
RP-CSPA	163.1	0.51	0.53	0.62	542341	**0.61**	0.69	0.69	3686	0.69	0.83	0.80
FM-CSPA	**145.9**	**0.55**	**0.61**	**0.74**	**529449**	0.58	**0.49**	**0.74**	**3011**	**0.71**	**0.84**	**0.85**
RS-HGPA	271.1	0.01	0.49	0.53	581715	0.14	0.47	0.67	3616	0.69	0.83	0.80
RP-HGPA	202.3	0.24	0.49	0.52	567975	0.25	0.48	0.67	3624	0.70	0.84	0.81
FM-HGPA	**202.3**	**0.68**	**0.50**	**0.54**	**525676**	**0.29**	**0.49**	**0.69**	**3523**	**0.72**	**0.85**	**0.82**
RS-MLCA	171.5	0.02	0.50	0.54	570900	0.31	0.48	0.67	3619	0.72	0.85	0.82
RP-MLCA	184.0	**0.62**	0.54	0.65	547288	0.68	0.48	0.68	3608	0.73	0.86	0.83
FM-MLCA	**143.3**	0.59	0.61	**0.77**	**522337**	**0.50**	**0.49**	**0.71**	**3575**	**0.75**	**0.87**	**0.84**
KM-CSPA	170.6	0.47	0.56	0.68	614645	0.46	0.47	0.70	3798	0.59	0.81	0.70
KM-HGPA	202.3	0.47	0.49	0.54	645685	0.27	0.48	0.69	3705	0.65	0.84	0.77
KM-MCLA	149.9	0.48	**0.62**	0.71	608945	0.47	0.49	0.69	3755	0.61	0.85	0.71

The results of ensemble clustering with FastMap projection were also better than those with other two methods under ARI evaluation in all data sets. The majority best results were also obtained with FastMap method, except for one case of GLI-85. However, the difference was not very significant.

Under CP and NMI evaluations, the ensemble clustering with FastMap projection also outperformed the ensemble clusterings with other two methods in most data sets. The majority best results also occurred in the FastMap method. These results demonstrated that the FastMap projection for component data set generation improved the performance of ensemble clustering of high dimensional data.

4.4 Comparisons of FastMap Projection vs. Random Sampling and Random Projection

We used the three component data generation methods: random sampling, random projection and FastMap projection to generate component data sets from

3 real world data sets. The characteristics of these data sets are presented in Table 1 in the next section. We computed two measures on the component data sets. Table 3 shows the comparisons of three methods in intrinsic dimensionality. The component data sets are in three different dimensions. We can see that the intrinsic dimensionality values in FM columns are smaller than the values in RP and RS columns. These results indicate that FastMap can generate better component data sets than random sampling and random projection. One exception is data set **GLI85** which is a fat Microarray data with only 85 objects but 22283 features. Although FastMap projection was a little worse than random projection in intrinsic dimensionality, we shown in the previous section that the ensemble clustering results of FastMap projection in this data set are still better than the results from other two methods.

Table 3. Intrinsic dimensionality.

Data sets	5-dimensional space			10-dimensional space			15-dimensional space		
-	FM	RP	RS	FM	RP	RS	FM	RP	RS
Internet Ad	3E-05	5E-05	0.526	2E-05	4E-05	0.485	1E-05	4E-04	0.390
GLI85	5E-11	2E-11	5E-09	5E-11	1E-11	2E-09	9E-11	1E-10	1E-10
Orlraws10P	1E-06	5E-05	1E-03	2E-06	9E-05	8E-03	1E-06	1E-04	7E-03

Table 4 shows the comparisons of three methods in distance preservation. The values of distance preservation in FM columns are much smaller than those in RP and RS columns. These results demonstrate that FastMap projection has better distance preservations than other two methods.

Table 4. Distance preservation.

Data sets	5-dimensional space			10-dimensional space			15-dimensional space		
-	FM	RP	RS	FM	RP	RS	FM	RP	RS
Internet Ad	0.681	0.739	1.812	0.690	0.901	1.001	0.671	0.719	1.009
GLI85	0.596	0.639	0.960	0.539	0.590	0.921	0.447	0.46	0.931
Orlraws10P	0.479	0.519	1.238	0.394	0.469	1.108	0.359	0.493	0.998

5 Conclusions

In this paper, we have presented the FastMap projection method to generate low dimensional component data sets for ensemble clustering. We have analyzed FastMap projection, random sampling and random projection and demonstrated that FastMap projection can better preserve the clustering structure of the original data than other two methods. Because of this property, the ensemble clustering with FastMap projection outperformed ensemble clusterings with other two methods in experiments on three real world high dimensional data sets. Beside better performance, another advantage of FastMap is that it is efficient in handling big data and flexible in component data generation.

Acknowledgment. This research is supported by Shenzhen New Industry Development Fund under Grant No. JC201005270342A.

References

1. Law, M., Topchy, A., Jain, A.: Multiobjective data clustering. In: Proceedings of CVPR, pp. 424–430 (2004)
2. Chávez, E., Navarro, G.: A probabilistic spell for the curse of dimensionality. In: Buchsbaum, A.L., Snoeyink, J. (eds.) ALENEX 2001. LNCS, vol. 2153, pp. 147–160. Springer, Heidelberg (2001)
3. Bingham, E., Mannila, H.: Random projection in dimensionality reduction: applications to image and text data. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 245–250 (2001)
4. Fern, X.Z., Brodley, C.E.: Random projection for high dimensional data clustering: a cluster ensemble approach. In: International Conference on Machine Learning (2003)
5. Faloutsos, C., Lin, K.: Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In: Proceedings of ACM-SIGMOD, pp. 163–174 (1995)
6. Xue, H., Chen, S., Yang, Q.: Discriminatively regularized least-squares classification. Pattern Recognit. **42**, 93–104 (2009)
7. Dasgupta, S.: Experiments with random projection. In: Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference, pp. 143–151 (2000)
8. Domeniconi, C., Al-Razgan, M.: Weighted cluster ensembles: methods and analysis. ACM Trans. KDD **2**, 1–40 (2009)
9. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. **20**, 359–392 (1998)
10. Kriegel, H., Kroger, P., Zimek, A.: Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering and correlation clustering. ACM Trans. KDD **3**, 1–58 (2009)
11. Aswani Kumar, C.: Reducing data dimensionality using random projections and fuzzy k-means clustering. Int. J. Intell. Comput. Cybern. **4**, 353–365 (2011)
12. Karypis, G., Aggarwal, R., Kumar, V., Shekhar, S.: Multilevel hypergraph partitioning: applications in vlsi domain. In: Proceedings of Conference on Design and Automation (1997)
13. Kuncheva, L., Vetrov, D.: Evaluation of stability of k-means cluster ensembles with respect to random initialization. IEEE Trans. PAMI **28**, 1798–1808 (2006)
14. Kuncheva, L.I., Hadjitodorov, S.T.: Using diversity in cluster ensembles. In: IEEE International Conference on Systems, pp. 1214–1219 (2004)
15. Strehl, A., Ghosh, J.: Cluster ensembles: a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**, 583–617 (2002)
16. Zhang, L., Mahdavi, M., Jin, R., Yang, T., Zhu, S.: Recovering optimal solution by dual random projection. In: JMLR: Workshop and Conference Proceedings, vol. 30, pp. 1–23 (2012)

A General Framework for Parallel Unary Operations on ZDDs

Shogo Takeuchi[1](✉), Takahisa Toda[2], and Shin-ichi Minato[1,3]

[1] ERATO MINATO Discrete Structure Manipulation System Project,
Japan Science and Technology Agency, Hokkaido University,
Sapporo 060-0814, Japan
takeuchi@erato.ist.hokudai.ac.jp

[2] Graduate School of Information Systems,
The University of Electro-Communications, Chofu, Tokyo 182-8585, Japan
takahisa.toda@is.uec.ac.jp

[3] Graduate School of Information Science and Technology,
Hokkaido University, Sapporo 060-0814, Japan
minato@ist.hokudai.ac.jp

Abstract. A zero-suppressed binary decision diagram is a compressed data structure that represents families of sets. There are various basic operations to manipulate families of sets over ZDDs such as union, intersection, and difference. They can be efficiently computed without decompressing ZDDs. Among them, there are many important unary operations such as computing the ZDD for all extremal sets (maximal sets or minimal sets) from an input ZDD. Unary operations are useful in various fields such as constraint programming, data mining, and artificial intelligence. Therefore, they must be efficiently computed. In this paper, we propose a general framework for parallel unary operations on ZDDs. We analyze the computational complexity and evaluate the effectiveness of our method by performing computational experiments.

Keywords: Parallelization · Zero-suppressed binary decision diagram · Compression

1 Introduction

Computational problems that are actually required to be solved are becoming increasingly larger in scale and more complicated. Many such problems tend to require an exhaustive computation such as considering all possible cases and often turn out to be hard to solve within a realistic time.

To overcome the computational difficulty caused by combinatorial explosion, a compressed data representation, called a zero-suppressed binary decision diagram (ZDD), has recently come into use. A ZDD, a variant of a binary decision diagram (BDD), is considered as an efficient data representation for a set family. Coudert [1] proposed a framework to solve various set optimization problems

Takahisa Toda – This work was supported by JSPS KAKENHI Grant Number 26870011.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 494–503, 2014.
DOI: 10.1007/978-3-319-13186-3_44

using ZDDs. He demonstrated that various problems could be solved by combining a small number of fundamental set operations and that such operations could be elegantly implemented on ZDDs. A ZDD operation is to construct the output ZDD from possibly multiple input ZDDs, which corresponds to some set operation, without explicit decompression, and it often effectively manipulates large-scale set families. Knuth [2] treats ZDDs in detail in his popular textbook. BDD and ZDD-based techniques have recently been proposed for solving large-scale problems in real life such as exact evaluation of network reliability [3,4], optimal configuration in a distribution network [5], and frequent pattern mining [6–9].

Most studies concerning parallelizing BDD operations were done in the 1990s. Their main interest was in how to manipulate huge BDDs efficiently on a disk that are too large to store on a main memory. To achieve this, many practical techniques have been presented, such as exploiting disk cache in executing BDD operations on the basis of breadth-first search. Since then, although much larger memory and multi-cores have become available, there seems to have been almost no recent research concerning multi-thread parallelization on a shared memory environment. In 2013, Dijk et al. [10] extended an existing lockless hash table and implemented a uniquetable, garbage collection, and operation cache. Furthermore, they developed a multi-core BDD package on the basis of load balancing using a work-stealing framework (Wool), and applied it to symbolic model checking. Elbayoumi et al. [11] developed a cache friendly BDD package using the hopscotch hash table, which is a state-of-the-art hash table presented by Herlihy et al. in 2008.

One main difficulty in parallelizing BDD (and also ZDD) operations lies in task distribution. That is, in graph processing, it is usually unpredictable where and how much computational resources are consumed, and thus load balancing becomes a hard task. Therefore, a naive method is likely to result in performance degradation.

In this paper, we present a general framework for parallelizing unary operations of ZDDs, where a unary operation is the one that a single ZDD is given as its input. Since this framework is straightforward to apply to BDDs, we leave out a description of it for want of space. Our basic idea is as follows. Although parallelization of BDD operations generally presents difficulty in task distribution, by restricting such operations to be unary, we are able to realize task distribution that is expected to be moderately balanced without an external framework such as Wool. There are many practically important instances of unary operations such as the MIN and the MAX operations that compute all minimal and maximal sets from a given set family, respectively. As an application example, the MIN operation plays an essential role in generating all minimal hitting sets, which is a well-known important generation problem. A recently proposed method [12] uses a special version of the MIN operation, which receives the BDD for hitting sets and outputs the ZDD for all minimal hitting sets. In this paper, we evaluate our framework by measuring the performance of parallel MIN operation and SUP operation, which outputs all supersets for a set family on a fixed ground set.

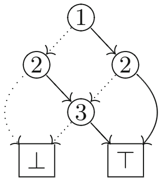
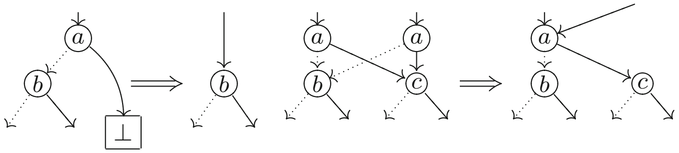


Fig. 1. ZDD for $\{\{2,3\},\{1,3\},\{1,2\}\}$



(a) Node elimination

(b) Node sharing

Fig. 2. Reduction rules for ZDDs

2 Zero-Suppressed Decision Diagrams

A *zero-suppressed binary decision diagram* (*ZDD*) is a graphical representation for set families. Figure 1 shows an example of a ZDD. Exactly one node has indegree 0, which is called the *root* and displayed at the top. Each internal node f has a label and exactly two children, which are indicated by the three fields $\mathrm{V}(f)$, $\mathrm{LO}(f)$, $\mathrm{HI}(f)$ associated with f. Each node has an element in a ground set V as its label. The children indicated by $\mathrm{LO}(f)$ and $\mathrm{HI}(f)$ are called the LO *child* and HI *child* of f, respectively. The arc to a LO child is called a LO *arc* and illustrated by a dashed arrow, while the arc to a HI child is called a HI *arc* and illustrated by a solid arrow. There are only two terminal nodes, denoted by $\top$ and $\bot$.

ZDDs satisfy the following two conditions. They must be *ordered*: if a node u points to an internal node v, then $\mathrm{V}(u) < \mathrm{V}(v)$. They must be *reduced*: the following reduction operation rules cannot be applied any further.

1. If there is an internal node u whose HI arc points to $\bot$, then redirect all the incoming arcs of u to the LO child, and then eliminate u (Fig. 2(a)).
2. If there are two internal nodes u and v such that the subgraphs rooted by them are equivalent, then share them (Fig. 2(b)).

We can understand ZDDs as follows. Given a ZDD, each path from the root to $\top$ corresponds to a set U in such a way that $k \in U$ if the HI arc of a node with label k is selected; otherwise, k is excluded. For example, in Fig. 1, the paths ① ⇢ ② → ③ → ⊤, ① → ② ⇢ ③ → ⊤, and ① → ② → ⊤ correspond to $\{2,3\}$, $\{1,3\}$, and $\{1,2\}$, respectively. Note that although the node ③ does not appear in the latter path, according to the node elimination rule, the HI arc of ③ must point to $\bot$, and thus the element 3 is excluded.

It is known (see for example [2,13]) that if the order in V is fixed, then set families on V correspond in a one-to-one way to ZDDs whose labels are taken from V. This means that different ZDDs represent different set families, and every set family has its own ZDD representation. For efficiency, ZDD nodes are maintained by a hash table, called a *uniquetable*, so that for any triple (k, lo, hi) of a node label and two ZDD nodes, there is a unique ZDD node f with $\mathrm{V}(f) = k$, $\mathrm{LO}(f) = lo$, and $\mathrm{HI}(f) = hi$. Given a triple (k, lo, hi), the function getnode returns an associated node in the uniquetable if it exists; otherwise, a new node f is created such that $\mathrm{V}(f) = k$, $\mathrm{LO}(f) = lo$, and $\mathrm{HI}(f) = hi$; f is registered to the uniquetable, and f is then returned. A uniquetable guarantees that two nodes are different if and only if the subgraphs rooted by them represent different set families. Thus, for example, equivalence checking of set families can be done in constant time.

For any node in a ZDD, the subgraph rooted by the node is a ZDD. Thus, if there is no confusion, we often identify nodes with the ZDDs rooted by them. The number of internal nodes in a ZDD f is denoted by $|f|$ and called the *size* of f.

An advantage of ZDDs is that thanks to the reduction rules, ZDDs tend to achieve a high compression efficiency and furthermore provide various basic operations to manipulate set families such as intersection, union, and difference. Such operations are performed with respect to the recursive structure of a ZDD, and the computation complexity is proportional to the product of input ZDD sizes [14]. Thus, by applying a sequence of such operations, we can quickly obtain a ZDD that represents a desired set family, as long as a good compression efficiency is retained during the computation. Therefore, algorithms must be appropriately designed for such operations so that ZDD sizes do not greatly increase.

3 A General Framework for Parallel Unary Operations

In this section, we introduce a lock-free uniquetable and propose a general framework for parallel unary operations.

3.1 Lock-Free Uniquetable

A standard way to coordinate concurrent accesses to shared objects is to use locks. However, when too many threads try to access a single object at the same time, a sequential bottleneck is likely to occur. Lock-free techniques resolve this problem by using atomic operations instead of locks, since atomic operations allow threads to read and write a value to memory in one indivisible hardware step [15].

This paper uses a lock-free uniquetable based on a chained hash table, where hash table entries correspond to ZDD nodes, and the NX field of a ZDD node is a reference to the next node. Figure 1 shows a getnode operation that allows threads to access a uniquetable simultaneously. This implementation uses an atomic operation compare&swap, where $\mathrm{compare\&swap}\,(\mathrm{NX}\,(t)\,, \mathrm{NULL}, new)$ substitutes *new* for NX field and returns 1 if the NX field of the current node t is NULL; otherwise, it returns 0 without substitution: note that these operations are safely executed in one indivisible hardware step. For simplicity, each thread has its own freelist of ZDD nodes, and our uniquetable does not support removal of ZDD nodes. If one wants to reuse ZDD nodes that are no longer in use, the uniquetable needs to be modified so as to enable removal of nodes, which makes implementation more complex (see [10]).

Algorithm 1. Search a ZDD node with label k, LO child lo, and HI child hi. If not found, insert a node with these fields. This function should be computed by the tid-th thread.

```
function getnode_para($tid, k, lo, hi$)
    if $hi = \bot$ then
        return $lo$;
    end if
    Get a node $new$ from the freelist of the $tid$-th thread;
    $\mathrm{V}(new) \leftarrow k$; $\mathrm{LO}(new) \leftarrow lo$; $\mathrm{HI}(new) \leftarrow hi$; $\mathrm{NX}\,(new) \leftarrow \mathrm{NULL}$;
    $t \leftarrow$ the head node of a bucket determined by the hash value of $(k, lo, hi)$;
    while compare&swap $(\mathrm{NX}\,(t), \mathrm{NULL}, new) = 0$ do
        $t \leftarrow \mathrm{NX}\,(t)$;
        if $\mathrm{V}(t) = k$ and $\mathrm{LO}(t) = lo$ and $\mathrm{HI}(t) = hi$ then
            Put back $new$ to the freelist; return $t$;
        end if
    end while
    return $new$;
end function
```

3.2 Algorithms

Our framework can parallelize any unary operation that is recursively defined on the structure of an input ZDD as follows.

```
function $\mathrm{unaryop}_{(op_\bot, op_\top, op_{\mathrm{LO}}, op_{\mathrm{HI}})}(f)$
    if $f$ is a terminal node then
        return $op_f$;
    end if
    $l \leftarrow \mathrm{unaryop}_{(op_\bot, op_\top, op_{\mathrm{LO}}, op_{\mathrm{HI}})}\,(\mathrm{LO}(f))$;
    $h \leftarrow \mathrm{unaryop}_{(op_\bot, op_\top, op_{\mathrm{LO}}, op_{\mathrm{HI}})}\,(\mathrm{HI}(f))$;
    return getnode $(\mathrm{V}(f), op_{\mathrm{LO}}(l, h), op_{\mathrm{HI}}(l, h))$;
end function
```

That is, if an input ZDD f is a terminal node, then the result is a terminal node determined by a constant function $op_\bot$ for $f = \bot$ and $op_\top$ for $f = \top$. Otherwise, the result is computed by applying op_{LO} and op_{HI} to the results of its children $\mathrm{LO}(f)$ and $\mathrm{HI}(f)$, where op_{LO} and op_{HI} are any binary operations. This constraint is not very strict, and one can easily find unary operations of the form above.

Our framework consists of the following two parts.

1. Sort internal nodes of an input ZDD f in ascending order of *maximum path length* (i.e. the maximum length of a path to $\top$). This part is done with a single thread.
2. Let multiple threads mutually exclusively acquire and process nodes in sorted order in such a way that after each thread scans all nodes of the same maximum path length, it waits until the other nodes finish.

In the latter part above, the processing for each node g, which is described in Algorithm 3, is to compute the result of applying a unary operation to g in a bottom-up fashion, and then to set the result to its auxiliary field $\mathrm{AUX}\,(g)$. Thus, after all nodes are examined, the AUX field of the root node holds the final output.

Algorithm 2. The function csort sets internal nodes of a ZDD f to an array A in ascending order of maximum path lengths with the aid of the subroutine maxlen. The AUX field of each node holds its maximum path length.

```
function csort($f, A$)
    Clear the AUX fields of all nodes in $f$;
    $\mathrm{AUX}\,(\bot) \leftarrow \mathrm{AUX}\,(\top) \leftarrow 0$;
    $\mathrm{maxlen}\,(f)$;
    Set all internal nodes to $A$ while traversing $f$.
    Sort $A$ by counting sort with respect to maximum path length.
end function

function maxlen($f$)
    if $\mathrm{AUX}\,(f)$ is not set, and $f$ is an internal node then
        $m_0 \leftarrow \mathrm{maxlen}\,(\mathrm{LO}(f))$; $m_1 \leftarrow \mathrm{maxlen}\,(\mathrm{HI}(f))$;
        $\mathrm{AUX}\,(f) \leftarrow \max\{m_0, m_1\} + 1$;
    end if
end function
```

The function csort defined in Algorithm 2 computes the former part of our framework. First, it computes the maximum path length for each internal node and sets its value to the AUX field. The traversal of all nodes in a ZDD f requires $O(|f|)$ time, as does the subroutine maxlen. Let m be the maximum path length of the root node of f. Since $m \leq |f|$, the counting sort part requires $O(|f|)$ time. Therefore, the total time for csort is $O(|f|)$. An extra space is required essentially in the traversal of f and in counting sort, which is clearly $O(m)$ in total. From the argument above, we have the following proposition.

Proposition 1. *Algorithm 2 can be implemented to run in time proportional to an input ZDD size. The required extra space is proportional to the maximum path length of the root node.*

The function unaryop_thread defined in Algorithm 3 is in charge of the latter part of our framework. For each internal node g, the resulting ZDD is stored in the AUX field of g. We can safely use the AUX field of g, since the maximum path length for g is not needed after that. When we set to work on g, the results for their children $\mathrm{LO}(g)$ and $\mathrm{HI}(g)$ must already be computed. This is guaranteed in our framework, because we scan nodes in ascending order of maximum path length, and from the following proposition, it follows that each node is examined after the works for its children are over.

Algorithm 3. Compute the results of applying the unary operation characterized by $op_\bot, op_\top, op_{\mathrm{LO}}, op_{\mathrm{HI}}$ to nodes in the section of an array A from *curr* to *end*, which initially hold the start and end positions of nodes with the same maximum path length, respectively. The variable *curr* is shared with other threads.

```
function unaryop_thread($op_\bot, op_\top, op_{\mathrm{LO}}, op_{\mathrm{HI}}, tid, A, curr, end$)
    AUX ($\bot$) $\leftarrow op_\bot$; AUX ($\top$) $\leftarrow op_\top$;
    loop
        $i \leftarrow$ fetch&add ($curr, 1$);
        if $i > end$ then
            return NULL;
        end if
        $g \leftarrow A[i]$; $l \leftarrow$ AUX (LO($g$)); $h \leftarrow$ AUX (HI($g$));
        AUX ($g$) $\leftarrow$ getnode_para ($tid$, V($g$) , $op_{\mathrm{LO}}(l,h)$, $op_{\mathrm{HI}}(l,h)$);
    end loop
end function
```

Proposition 2. *If a node g has maximum path length i, then its children have a maximum path length less than i.*

Proof. Assume that a child of g has maximum path length j $(\geq i)$. This implies that there is a path longer than j starting from g and leading to $\top$. >From the maximality of i, we have $j < i$, which is a contradiction.

The function fetch&add is an atomic operation, and fetch&add $(curr, 1)$ executes in one indivisible hardware step that it first returns the current position *curr* and then increments *curr* by one. Thus, different threads cannot acquire an identical ZDD node. We can safely obtain ZDD nodes with getnode_para. Therefore, nodes in the same section can be completely independently processed by multiple threads, although since processing costs vary depending on nodes, threads that finish earlier must wait for the other threads.

Processing by multiple threads is better than that by a single thread in the sense that extra costs for parallel processing are the overhead caused by atomic operations and the cost by Algorithm 2; the former cost can be ignored, while according to Proposition 1, the latter cost cannot be a computational bottleneck if the computation requires much more than an input size. In particular,

Table 1. Data sizes

	HIT		SUP	
	$\lvert BDD\rvert$	#Itemsets	$\lvert BDD\rvert$	#Itemsets
p6_1000	25,640,936	2,251,757,096,338,520	25,640,936	42,717,346,724
win25600	1,250,409	overflow	1,250,409	overflow
lose12800	2,200,193	overflow	2,200,193	overflow
bms2_10	2,386,383	overflow	2,386,383	8,880,670
ac_30k	350,954	overflow	350,954	31,828,666

a significant effect of parallel processing will be expected if processing costs of nodes are distributed evenly. In another scenario, if the cost of a computation requires only an input size, then unfortunately our method will not be a good choice.

4 Experiments

To evaluate performance of the proposed framework, we conducted computational experiments.

4.1 Experimental Settings

We performed experiments on a computer that consists of four octa-core 2.67 GHz Intel Xeon E7-8837 processors (i.e., 32 CPU cores in total) and 1.5 TB DDR3 memory shared among cores. We implemented all the algorithms in C and compiled with the version 4.3.4 of GNU C compiler. We used the atomic function offered by GNU C compiler.

To make input data, we used p6_1000, win25600, lose12800, bms2_10 and ac_30k from the Hypergraph Dualization Repository[1]. We computed the BDDs that represent hitting sets and supersets of above datasets.

Table 1 summarizes the sizes of data. The sizes of both BDDs and uncompressed data are the number of BDD nodes and the number of item sets, respectively. "Overflow" represents the number of item sets larger than 2^{64}. We can see BDDs represents the very large data in a very compact way.

4.2 Experimental Results

We computed all minimal sets for each input dataset. Table 2 shows the execution time, and Fig. 3 shows the speedups of unary operation when the number of CPU cores (shown as "#Threads") varies.

The proposed method with multi-cores computes minimal sets more quickly than that with a single core. For p6_1000, the proposed method scaled well, and speedup is around four with eight cores in both the hitting set and the superset. On the other hand, the speedups saturated and resulted in around 1.5 for datasets bms2_10 and ac_30k. One reason for inefficient parallel computation is that most execution time is short (less than 10 seconds) except for p6_1000 and bms2_10. In such a case, both start-up and termination overhead becomes larger in the total execution time. When near the root node, the number of nodes to process decreases and execution time per node increases, our method cannot scale well even though execution time is long enough.

[1] http://research.nii.ac.jp/~uno/dualization.html, accessed on 14 Jan. 2014.

Table 2. Experimental results (sec.)

# Threads		1	2	4	8
HIT	p6_1000	261.86	152.64	94.62	64.69
	win25600	5.46	3.7	2.63	1.8
	lose12800	12.27	8.34	6.1	4.2
	bms2_10	187.02	141.37	117.03	113.16
	ac_30k	1.48	1.21	1.04	0.93
SUP	p6_1000	46.69	29.95	19.98	15.06
	win25600	2.42	1.66	1.17	0.8
	lose12800	3.62	2.9	2.02	1.5
	bms2_10	4.71	3.68	3.2	3.44
	ac_30k	2.52	2.27	1.88	1.84

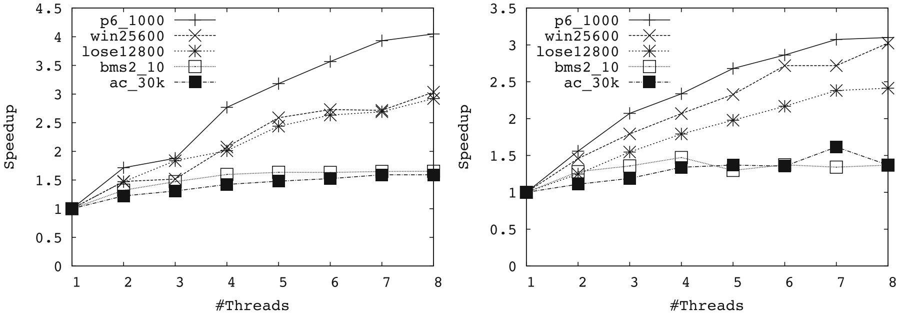


Fig. 3. Speed-up (left : Hitting set, right: Superset)

5 Conclusion

In this paper, we proposed a general framework for parallel unary operations on ZDDs. To achieve efficient parallelization, we focused on unary operations and shared memory parallelization. We analyzed the computational complexity and evaluated the effectiveness of our method by computational experiments.

References

1. Coudert, O.: Solving graph optimization problems with ZBDDs. In: The 1997 European Conference on Design and Test, Paris, France, pp. 224–228, March 1997
2. Knuth, D.: The Art of Computer Programming, vol. 4a. Addison-Wesley Professional, New Jersey (2011)
3. Sekine, K., Imai, H.: Counting the number of paths in a graph via BDDs. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. E80-A, 682–688 (1997)
4. Hardy, G., Lucet, C., Limnios, N.: K-terminal network reliability measures with binary decision diagrams. IEEE Trans. Reliab. **56**(3), 506–515 (2007)

5. Inoue, T., Takano, K., Watanabe, T., Kawahara, J., Yoshinaka, R., Kishimoto, A., Tsuda, K., Minato, S.I., Hayashi, Y.: Distribution loss minimization with guaranteed error bound. IEEE Trans. Smart Grid **5**(1), 102–111 (2014)
6. Minato, S., Arimura, H.: Frequent closed item set mining based on zero-suppressed BDDs. Trans. Jpn. Soc. Artif. Intell. **22**, 165–172 (2007)
7. Minato, S., Arimura, H.: Frequent pattern mining and knowledge indexing based on zero-suppressed BDDs. In: Džeroski, S., Struyf, J. (eds.) KDID 2006. LNCS, vol. 4747, pp. 152–169. Springer, Heidelberg (2007)
8. Minato, S., Uno, T., Arimura, H.: LCM over ZBDDs: fast generation of very large-scale frequent itemsets using a compact graph-based representation. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 234–246. Springer, Heidelberg (2008)
9. Loekito, E., Bailey, J., Pei, J.: A binary decision diagram based approach for mining frequent subsequences. Knowl. Inf. Syst. **24**(2), 235–268 (2010)
10. van Dijk, T., Laarman, A., van de Pol, J.: Multi-core BDD operations for symbolic reachability. Electron. Notes Theor. Comput. Sci. **296**, 127–143 (2013)
11. Elbayoumi, M., Hsiao, M.S., ElNainay, M.: A novel concurrent cache-friendly binary decision diagram construction for multi-core platforms. In: Design, Automation Test in Europe Conference Exhibition (DATE), pp. 1427–1430, March 2013
12. Toda, T.: Hypergraph transversal computation with binary decision diagrams. In: 12th International Symposium on Experimental Algorithms, Rome, Italy, June 2013
13. Minato, S.: Zero-suppressed BDDs for set manipulation in combinatorial problems. In: 30th ACM/IEEE Design Autiomation Conference (DAC-93), Dallas, Texas, USA, pp. 272–277, Jun 1993
14. Schröer, O., Wegener, I.: The theory of zero-suppressed BDDs and the number of knight's tours. Formal Meth. Syst. Des. **13**(3), 235–253 (1998)
15. Herlihy, M., Shavit, N.: The Art of Multiprocessor Programming. Morgan Kaufmann Publishers Inc., San Francisco (2008)

On the Size of the Zero-Suppressed Binary Decision Diagram that Represents All the Subtrees in a Tree

Norihito Yasuda[1(✉)], Masaaki Nishino[2], and Shin-ichi Minato[1,3]

[1] JST ERATO Minato Discrete Structure Manipulation System Project,
Sapporo, Japan
yasuda@erato.ist.hokudai.ac.jp
[2] NTT Communication Science Laboratories, Kyoto, Japan
[3] Hokkaido University, Sapporo, Japan

Abstract. This paper presents a method of constructing a ZDD that represents all connected subtrees in the given tree and analyzes the size of the resulting ZDD. We show that the size of the ZDD is bounded by $O(nh)$ for a tree with n-nodes and h-height. Furthermore, by properly ordering the ZDD variables, we can further reduce the size to $O(n \log n)$, which is surprisingly small compared to represent at most $O(2^n)$ subtrees.

1 Introduction

In this paper we consider representing all subtrees in the given rooted tree using Zero-Suppressed Binary Decision Diagrams (ZDDs), a variant of Binary Decision Diagrams (BDDs) [1]. If we use a ZDD to represent subtrees, we can leverage the power of the decision diagrams to perform various operations on subtrees such as enumeration, set operations including intersection or union, and so on. Since each subtree can be seen as a set of vertices or edges and ZDD can, in many cases, efficiently represent a family of sets, however the number of subtrees can be extremely huge. For example, a n-node star, a tree that has $n-1$ leaves, has $2^{n-1}+n-1$ subtrees. We believe that no known works reveals that how efficiently (or how badly) ZDDs can represent this huge number of set.

In general, there are two approaches constructing ZDDs; first is the 'bottom up' approach that repeatedly uses Bryant's apply-algorithm [1]. Alternatively, the 'top-down' approach can be used to construct nodes from the root to terminals with a single pass. The latter approach is adopted, because the former approach requires an exponential number of steps. In top-down methods, a table or an array, also known as *mate*, is introduced to manage to check which nodes can be shared in the ZDD. Since not all vertices in the tree relate to decide each level of ZDD nodes, we only have to maintain *mate* entries for the set of vertices that have edges to decided vertices and undecided vertices. This set of vertices is called *frontier* [2] or *elimination front* [3,4].

We can roughly estimate BDD/ZDD size from the number of varieties of mate entries, and this tells us the size is $O(nh^2)$ for an n-vertices and h-height

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 504–510, 2014.
DOI: 10.1007/978-3-319-13186-3_45

rooted tree. By taking into consideration zero-suppression, the required number of nodes can be reduced. We will show that ZDD requires only $O(nh)$ nodes. Moreover, by properly ordering the tree edges, each of which is corresponded to a ZDD variable, we also show that the bound can be further tightened to $O(n \log n)$.

The resulting subtree ZDD has many applications. For example, we can easily construct an algorithm to construct a table of subtree sizes to their counts as follows:

	Subtree counts for Fig. 2										
# nodes	2	3	4	5	6	7	8	9	10	11	12,
count	11	15	22	30	38	43	41	31	17	6	1

	Subtree counts for Fig. 4													
# nodes	2	3	4	5	6	7	8	9	10	11	12	13	14	15.
count	14	19	26	38	52	71	94	114	116	94	60	28	8	1

This table can be obtained by using a *generating function* [2], and its time complexity is $O(n^2 \log n)$ for n-nodes tree. Another fascinating example is subtree pattern matching, such as jumbled pattern matching on trees [5]. Since subtree ZDD can compactly represent all subtrees, if another ZDD that represents any combination that matches the pattern, without regard to the tree structure, is also reasonably small, the straightforward approach of taking the intersection of these two ZDDs will be useful.

2 Top-Down Construction of the Subtree ZDD

We represent each subtree as a set of used edges and construct a ZDD whose variables correspond to every edge in the given rooted tree. Thus the total number of the ZDD variables will be equal to the size of the target tree minus 1. Note that we can also construct ZDDs whose variables correspond to tree vertices in the same way. Also note that when we use edges, the resulting ZDD does not contain trivial subtrees that consists of single vertices.

Following conventional methods, top-down construction determines ZDD nodes from the root to the terminal sink nodes with a single pass. Top-down construction can be designed by defining (1) the traversal order of the target graph, and (2) information associated with frontier vertices (the *mate*).

For traversal order, we adopt the preorder of the tree. Since the frontier is a set of vertices that have edges to decided vertices and undecided vertices as noted previously, the frontier of the target vertex x as a set of vertices that appears in the path from the root to the parent of x, with edges to undecided vertices. For example, let us suppose that the target tree is Fig. 1 and we represent each

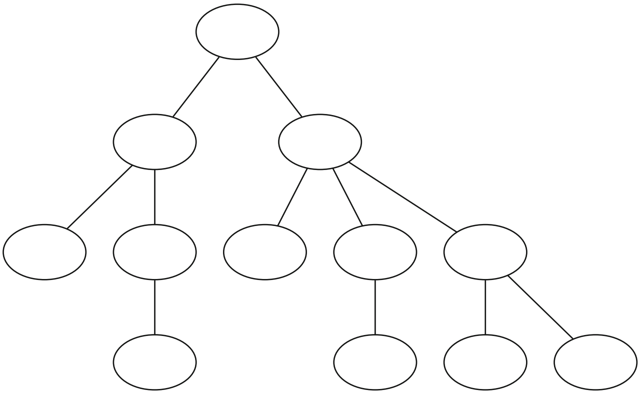


Fig. 1. Sample tree #1

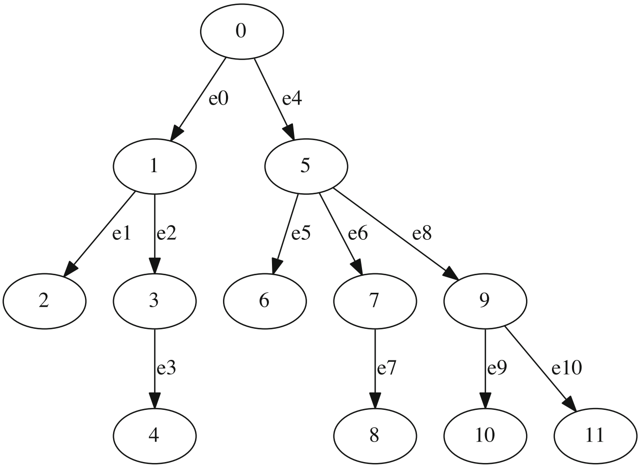


Fig. 2. Sample tree #1 labeled with our traversal order

subtree as a set of edges labeled as Fig. 2. In this example, when we define edge 1–2, the frontier will be $\{0,\ 1\}$, and $\{5\}$ for edge 5–7.

The role of the *mate* is to tell the following undecided nodes if they can consist subtrees or not, so we can represent this by a binary array of frontier length. Among the 0/1 patterns of this array, given that we are making subtrees, patterns in which vertices in the frontier are disconnected never appear. Thus, the patterns in the *mate* has to distinguish are $\frac{f(f-1)}{2}$ when the size of the frontier is f. This estimation can be applied both to BDDs and ZDDs. If we take zero-suppression and the true/false status of the target node into consideration, we can further tighten the estimation. When the target node is true, the above-mentioned variations in the *mate* are limited to such combinations that 1s (i.e. the edge is selected) must successively appear from the parent of x to make a subtree. All other variations will directly fall into the bottom sink, that

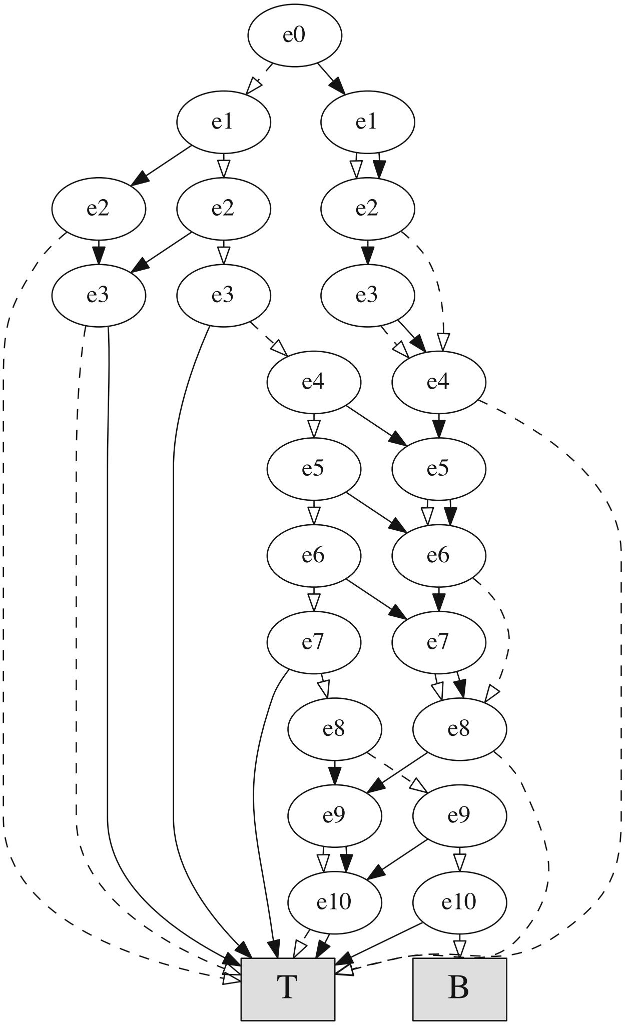


Fig. 3. The subtree ZDD for sample tree #1

is, zero-suppressed. As a result, at most f variations remain for each level of the ZDD.

Figure 3 depicts the resulting subtree ZDD for the tree in Fig. 2. Another example is shown in Fig. 4 and the resulting subtree ZDD in shown in Fig. 5.

3 Estimating the ZDD Size

Each subtree represented in the resulting ZDD is identified as a path from the ZDD's root to the top sink. Since each ZDD node has only two child nodes, to

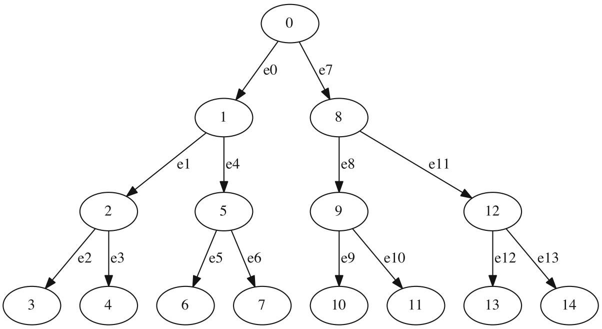


Fig. 4. Sample tree #2

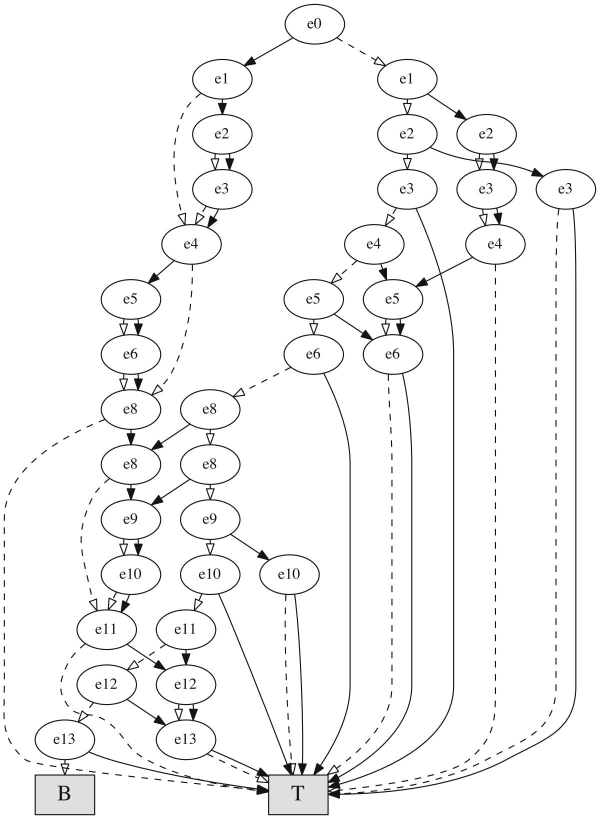


Fig. 5. The subtree ZDD for sample tree #2

convey information about connections from higher level variable to lower level, we require multiple nodes at the same level. This determines the ZDD's width, and this widths corresponds to the number of variations represented by *mate*. In our top-down construction, this width is equal to the length of frontier, and the length is at most the height of the tree. Therefore, we can estimate subtree ZDD size estimation by $O(nh)$.

In general, variable order impacts BDD size, so to reduce the size of the resulting ZDD, we introduce the following ordering.

Definition 1. *A right-heavier ordering is a special type of preordering that restricts sibling ordering such that a node with a larger number of descendants comes later.*

The order in Fig. 2 satisfies this right-heavier ordering.

Lemma 1. *Among all preordering configurations, right-heavier ordering gives the smallest sum of frontier size at every tree vertex of the given ZDD.*

Proof. Assume that vertex x in the tree has k descendants and we denote each subtree rooted at these k descendants as $T_1, ..., T_k$. Every frontier of nodes in $T_i (1 \leq i \leq k)$ can be divided into the following three parts:

1) from root node to the parent of x,
2) possibly x,
and 3) the descendants of x.

Every descendant of $T_i (1 \leq i \leq k)$, (1) are the same and (3) is irrelevant to the order of T_i. As for (2), if we define the ordering as $T_{O1}, ..., T_{Ok}$ for these k subtrees, the frontiers for the nodes within T_{Ok} are 0, and 1 for others, since vertices that do not have edges to undecided vertices, they are not included in the frontier. Thus, processing the subtree with the largest number of descendants minimizes total frontier size, and the right heavy ordering satisfies the condition. □

Theorem 1. *For any n-nodes tree, the frontier size never exceeds $\log_2 n$ if we use right-heavier ordering.*

Proof. Assume that the size of frontier S is greater than $\log_2 n$ at leaf vertex x. According to the definition of the frontier, S branches must exist from the root to the parent of x. In addition, since we are using right-heavier ordering, there must exist a subtree whose size is not smaller than the subtree that contain x on each branch.

We denote y_1 as the parent of leaf x, and y_2 as the parent of y_1. To avoid the condition where x itself becomes the heaviest subtree, there must be a subtree rooted at a sibling of x. Thus, the subtree rooted at $y1$ must be greater than 3. In the same way, there must be a subtree rooted at some sibling of $y1$. Thus, the size of the subtree rooted at $y2$ must be greater than $2 * |y1| + 1$. Here $|y_1|$ denotes the size of the subtree rooted at y_1. Recurrently, size of the subtree rooted at y_i satisfies $|y_{i+1}| \geq 2|y_i| + 1$. This lead to $y_S = 2^S - 1$.

This contradicts $S > \log_2 n$, because this means $2^S - 1 > 2^{\log_2 n} = n$. Therefore, $S \leq \log_2 n$. □

Corollary 1. *For any n-nodes tree, the number of nodes of a subtree ZDD is not greater than $n \log_2 n$.*

4 Conclusions

In this paper we presented a method of constructing a ZDD that represents all the subtrees in the given tree. We also showed that the size can be reduced to $O(n \log_2 n)$ by defining the variable order according to a special type of tree preorder. This size is incredibly compact compared to the number of subtrees store, which can be exponential as expressed by $O(2^n - 1)$.

References

1. Bryant, R.E.: Graph-based algorithms for boolean function manipulation. IEEE Trans. Comput. **100**(8), 677–691 (1986)
2. Knuth, D.E.: The Art of Computer Programming. Combinatorial Algorithms Part 1, vol. 4A. Addison-Wesley, Upper Saddle River (2011)
3. Sekine, K., Imai, H., Tani, S.: Computing the tutte polynomial of a graph of moderate size. In: Staples, J., Katoh, N., Eades, P., Moffat, A. (eds.) ISAAC 1995. LNCS, vol. 1004, pp. 224–233. Springer, Heidelberg (1995)
4. Sekine, K.: Counting the number of paths in a graph via bdds. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **80**(4), 682–688 (1997)
5. Gagie, T., Hermelin, D., Landau, G.M., Weimann, O.: Binary jumbled pattern matching on trees and tree-like structures. arXiv preprint arXiv:1301.6127 (2013)

Data Mining in Social Networks

A Unified Fusion Framework for Time-Related Rank in Threaded Discussion Communities

Qiang You(✉), Weiming Hu, Ou Wu, and Haiqiang Zuo

National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing 100190, China
{qyou,wmhu,wuou,hqzuo}@nlpr.ia.ac.cn

Abstract. We propose a unified fusion framework for time-related rank, applied to find valuable posts or recommend answers in threaded discussion communities. In our model, we simultaneously consider the special structure and semantics of threaded discussion communities. As for the structure, we construct a time-related rank model with respect to reply posts analysis and attain an initial rank result. Concurrently, we reconstruct semantic trees from raw statistical features (e.g. term frequency and document length) to latent semantics and topics. With a more robust similarity computation, we produce several semantic trees. For each tree, we again compute the time-related rank score and get a series of rank results. Finally, we fuse our results in the unified fusion framework incorporating quality measures to make a final decision. Our model can be easily extended when new features or models are added. Experimental results show that our model contributes satisfactory results.

Keywords: Fusion · Quality measure · Time-related rank · Information retrieval

1 Introduction

With the Internet growing prosperously, increasing web users would like to share their hobbies, experiences etc. on the Internet. As a result, many kinds of threaded discussion communities are booming and play a more and more important role in content contribution for the web. Benefit from the openness of many threaded discussion communities, we can access the content and get the reply structure of each thread without much difficulty. Unlike the World Wide Web with huge and heterogeneous information, a threaded discussion community always keeps its eye on one or a few domains, which provides a concentrated source to us to access information for the specific domain. Providing the hotness and the character for a stable information dispatcher, mining the threaded discussion communities and gaining valuable posts or users become a urgent task. Unlike traditional mining tasks which conduct knowledge discovery on normalized data in database, the threaded discussion mining aims to find valuable information in non-normalized posts which are proposed by the web users constantly.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 513–524, 2014.
DOI: 10.1007/978-3-319-13186-3_46

Researchers have heavily counted on the vector space model such as term frequency (TF) to represent a document, which is based on the hypothesis the document is represented as an unordered collection of words, neglecting grammar and word order. To summarize and extract the main idea of a corpus with many related documents (always, the corpus is modeled as a matrix called the term document matrix), a series models are adopted to choose and weight the terms in the corpus. The latent semantic indexing (LSI) [1] is a widespread method in information retrieval to find the relations between terms and concepts by transforming the vector space to a new orthogonal space, which behaves effective in many applications (e.g. [2,3]). The latent Dirichlet allocation (LDA) [4] is a generative probabilistic model for collections of discrete data, which assumes that each document is a mixture of several topics and that each word's creation is attributable to one of the document's topics. LDA is a three-level hierarchical Bayesian model, where a topic is draw from the multinomial distribution conjugated with a Dirichlet distribution prior, and each word is draw from a multinomial probability conditioned on the topic.

Those models we mentioned above try to understand the meaning of the text corpora only from one perspective of the document content. While on the web, especially user generated content (UGC) web, there exists rich meta data besides the content, such as time stamp when the user posts a message or even reply structure that shows who replies whom. The threaded discussion community is a typical kind of those webs with rich structure information. Providing the more extra structure information then a bag of discrete text data, we can rank posts according to their value or recommend answer to the given question.

Classical structure methods like PageRank [5] or HITS [6] have achieved great success in information retrieval. However, they are not quite suitable for the threaded discussions without explicit link structures. What is more, the threaded discussion community always varies instantly, where the users may produce many new posts even at one minute, which is not suitable to PageRank because it is liable to the stability of the whole web.

In this paper, we propose a time-related rank model which both considers the time stamps of each posts and the reply-to structure of the discussion thread. With semantic reconstruction, we easily fuse the content to the proposed rank model. The main contributions of this paper are summarized as follows:

- We construct a unified framework to fuse different models incorporating quality measures. Our framework is carried out in two steps. First we construct the quality measure model, then we use the result as a priori to the fusion model. Posit that the threaded discussions in one community are in the same knowledge domain, we choose a subset of threaded discussions to evaluate the quality of the models. Then we popularize the result to the whole threaded discussions in the same community.
- We propose a time-related rank model to alleviate the influence of the time factor when rank different posts with different time stamps, which is difficult for classical models such as PageRank to handle.

– We propose a method to reconstruct the structure of the posts in a thread according to their semantics. Thus, the structure and semantics of a thread with many posts can be easily adopted in our unified fusion framework.

The rest paper is organized as follows. Section 2 introduces the related work. Section 3 briefly introduces the characteristics of threaded discussion communities. Section 4 prepares the needed work including the time-related rank model and the semantic reconstruction. Section 5 gives a careful description of our unified framework which combines several semantic models together based on quality measures. Section 6 provides a thorough set of experiments on two real data sets collected from the apple discussion forum[1] and *Slashdot.org*[2]. We conclude the paper in Sect. 7.

2 Related Work

To the best of our knowledge, little previous research studies the time-related rank in threaded discussion communities. However, there are still a lot of work related to the threaded discussions mining, which can be mainly categorized into semantic models and structure models. As for the semantic models, with information extraction, [7] aimed at ranking answers for given questions in web forums. References [8,9] reconstruct the relationship among posts and threads based on the similarity of topics and semantics. Previous structure models such as PageRank [5] and HITS [6] are under the assumption that the whole web is stable in general, while the threaded discussion communities are not. FGrank [10] modifies PageRank to suitable to the forum pages by constructing page level link graph based on the topic hierarchy without considering the reply-to graph of the posts. Other than the separated models, [11] proposes a sparse coding approach to simultaneously modeling semantics and structure of threaded discussions. It uses the reply-to graph as ground truth and justifies the reply reconstruction of the post by content similarity. However, we believe that the reply-to relationships of the posts should fuse with the content to get a better result.

3 The Problem Setting

Recent years, with the development of the World Wide Web, a lot of UGC web communities have been arising. The threaded discussion community is a typical UGC web community that doesn't emphasize the function of social communication like Facebook or Twitter but supply a place to solve problems or share profound insights, such as mailing list, BBS or Q&A forums. The typical structure of a threaded discussion community is shown in Fig. 1. A threaded discussion community is constituted by a lot of threads in the same knowledge domain. The process of the development of a threaded discussion community can be treated

[1] https://discussions.apple.com/community/ipad

[2] https://slashdot.org

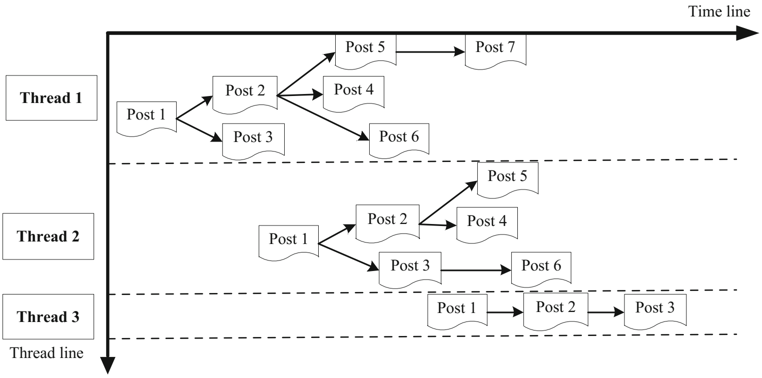


Fig. 1. A typical structure of a threaded discussion community

as the threads' creation and development, which is described as follows. First, one user releases the first post, which attracts a few users to discuss. Then, they propose posts one by one until a consensus is reached. With the creation of more and more threads, the threaded discussion community becomes mature. There are so many posts in a thread and so many threads in a threaded discussion community. Which posts are more valuable or have more insight then others and should be recommended to the other users? That is the main problem we should solve in this paper.

We assume that a threaded discussion community $\mathcal{C}$ is constituted by N threads, and the i-th thread D_i is represented as a directed graph $G_i(V, E, \mathbf{ts})$. The node $v \in V$ is associated with a post which can be modeled by TFIDF, LSI or LDA. There is a time stamp $ts_v \in \mathbf{ts}$ which stands for the moment when the post v is proposed. The edge $(u \rightarrow v) \in E$ exists between two node u, v if post u replies post v. Supposing that there is a metric function f mapping the post v to its value $f(v)$, we get the rank result just according to this value. PageRank is one of this metric function in ranking web pages according to their reputation. In threaded discussion communities, inspired by PageRank, we propose a time-related rank model which is more suitable for our problem.

4 The Preparation Work

Before the unified fusion process, we introduce the time-related rank model. Through semantic reconstruction with several existing vector space models, we easily fuse the content analysis in the rank model.

4.1 The Time-Related Rank Model

The rank model should consider three important factors in a threaded discussion with many posts if the post ranks high. (1) The post should be released *timely* in a thread. (2) The post should attract discussion posts *as many as possible.* With large amount of discussion, the post becomes focused and should also be recommended to the other users. (3) The post with many replied posts which should reply *immediately.* A post that attracts many users to discuss immediately shows the post is active in a short-term response. In conclusion, a post that is timely released and with large posts replying immediately should rank high.

Given a thread D represented by a directed graph $G(V, E, \mathbf{ts})$, we construct the model as follows. The weighted matrix W is calculated with the element $w(u, v) = K(ts_u, ts_v)$. We define a function $h(v) = H(ts_v)$ to depict the timeliness of post v in the thread. We treat the time-related rank (trr) score calculation of each node as an iterative procedure. In step t, the trr score of node v

$$trr^{(t)}(v) = h(v) \sum_{(u \to v) \in E} \frac{trr^{(t-1)}(u)}{w(u, v)} \tag{1}$$

We repeat the iterative procedure until divergence, and rank the posts in a thread with respect to the trr score.

4.2 Semantic Reconstruction

It is hard to analyze the semantics of each post individually because the post released by the users is short and sparse, which means the post itself has incomplete semantics and misses a large part of background knowledge. We reconstruct a semantic tree based on one vector space model from a thread with many posts where each node represents a post and near neighbors have similar semantics. Thus, the post is not individual in semantics with the help that the neighbors provide the context information in the semantic tree.

Given a thread D with m posts $\{L_i\}_{i=1}^{m}$, their time stamps $\{ts_i\}_{i=1}^{m}$ where $ts_i < ts_j$ if $i < j$ and the similarity measure function $S(L_i, L_j)$, we reconstruct the semantic tree through the following method. In our similarity computation, we define the similarity measure function as the weighted sum of two parts. The first part is a cosine similarity, and the second part is a similarity between two posts with respect to the post length. The parameter λ here weights the two parts.

$$S(L_i, L_j) = \lambda \frac{L_i L_j + \|L_i\| \, \|L_j\|}{2\|L_i\| \, \|L_j\|} + (1 - \lambda) \frac{2 \, \|L_i\| \, \|L_j\|}{\|L_i\|^2 + \|L_j\|^2} \tag{2}$$

As for post L_j, we choose one post as its predecessor from the ahead posts. The predecessor should have the most similarity with L_j.

$$L_* = \underset{L_i}{\arg} \max_{1 \le i \le j-1} S(L_i, L_j) \tag{3}$$

Let j decrement from m to 2, then the semantic tree is reconstructed.

After semantic reconstruction, we again calculate the *trr* score just as the reply structure analysis in a thread. As for the semantic tree and the reply structure graph, it is easy to tackle whatever “combine then rank” or “rank then combine”.

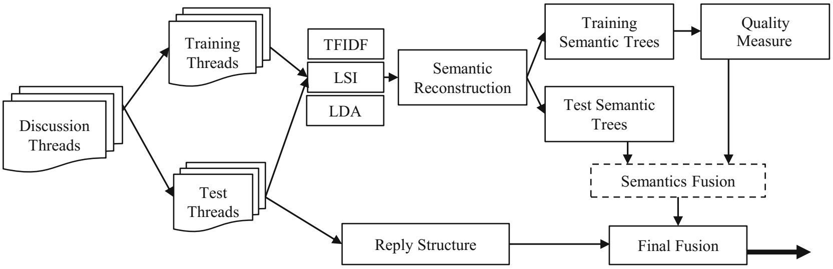


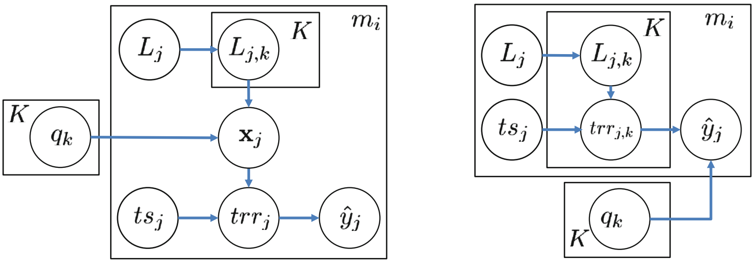
Fig. 2. The unified fusion framework based on the quality measures

5 The Unified Fusion Framework

In the introduction section, we have simply listed three vector space models: TFIDF, LSI and LDA. The first model is directed and contains many details of a document, while it may also include junks. The second model is a way to get the concepts or latent classes of a document, which is totally from the matrix decomposition of TFIDF, but may remove noise. LDA is a generative topic model which is widely used to cluster words with similar semantics. No one model can handle all the data sets. In our unified framework, we first construct a quality measure model to test how much the model is suitable for subset threads of the threaded discussion. The quality measure model is based on the assumption that in the same threaded discussion community, the same knowledge domain is adopted. For example, in the apple discussion forum, people are talking about the apple products. Second, we populate the quality of each model to the other threads. As Fig. 2 shows, our framework consists three main parts: semantic reconstruction, quality measure and fusion procedure. The first part has been described. Now we study the next two parts.

5.1 The Quality Measure for the Semantic Models

Our quality measure model is based on the assumption that all the discussion threads are in the same knowledge domain, which makes our model very suitable for the web communities that concentrate on a few central issues. Central discussions can produce profound insights easier than talking too many issues simultaneously. We randomly choose N_t threads from the whole threads. For the i-th thread D_i, there are m_i posts $\{L_j, ts_j, y_j\}_{j=1}^{m_i}$ in it where L_j is the content of the j-th post in thread D_i, ts_j is the time stamp of the post and y_j the label which stands for the rank or the score that the other users give. As for each model, there is a quality factor measuring the contribution to our result. Suppose that there are K semantic models. The quality vector is $\mathbf{q} = (q_1, .., q_k, .., q_K)$. With post L_j and its time stamp ts_j, after combining several semantic models, we get the output $\hat{y}_j = f(L_j, ts_j, \mathbf{q})$. Our quality measure becomes to solve the following optimization problem.



(a) Combine then rank

(b) Rank then combine

Fig. 3. The two strategies of quality measure for semantic models

$$\mathbf{q}_* = \arg\min_{\mathbf{q}} \sum_{i=1}^{N_t} \sum_{j=1}^{m_i} (\hat{y}_j - y_j)^2 \tag{4}$$

There are two strategies to handle the fusion of semantic models. They are *combine then rank* and *rank then combine*.

Combine then Rank. The strategy is a pre-combination which combines the models of the post with the quality factors, reconstructs one semantic tree and calculates trr score at last. As Fig. 3a shows, we obtain the fusion representation $\mathbf{x}_j$ of the post L_j by merging different representations into a new vector.

$$\mathbf{x}_j = (q_1 L_{j,1}, .., q_k L_{j,k}, .., q_K L_{j,K}) \tag{5}$$

Our semantic reconstruction is based on the new fusion representation. Given the reconstructed semantic tree, we calculate the trr score $trr(\{\mathbf{x}_j, ts_j\})$ and then translate it into output $\hat{y}_j = T\left(trr(\{\mathbf{x}_j, ts_j\})\right)$. The translation function $T(.)$ maps the trr score to rank or the mark the other users give.

Rank then Combine. The strategy first reconstructs each semantic tree on each model, then calculates trr score respectively, finally combines the trr scores in one score with the quality vector. As Fig. 3b shows, the output is

$$\hat{y}_j = T\left(\sum_{k=1}^{K} q_k trr(\{L_{j,k}, ts_k\})\right) \tag{6}$$

5.2 The Final Fusion

The reply structure of a thread and the semantics of the posts in the thread should be both considered in our problem. The posts those with much value and should be recommended to the other users must have at least two characteristics. (1) The posts should be ranked high with respect to the trr score in the reply structures, which suggests that the posts have much value in the eye of the users who have read the posts and actively participated in the discussion. (2) The posts should be ranked high in the semantic tree. The semantic tree is based on the similarity measures between posts. Those posts which are ranked high in the semantic tree suggest that they are more similar to the thoughts of the other users.

In the framework of the time-related rank model, we can easily combine the results of the two important factors:

$$trr = \alpha trr_{st} + (1 - \alpha)trr_{se} \tag{7}$$

where trr_{st} represents the trr score from the reply structure of the thread, and trr_{se} stands for the trr score based on the semantic reconstruction. The parameter α can be acquired by training the subset threads used in the quality measure. Then we rank each post in the thread according to the trr score.

6 Experiments

We collect two kinds of data sets over a period of time by a web crawler designed for the threaded discussion communities. One is from the iPad Q&A board in the apple discussion forum, the other is from the technique community *Slashdot.org*. These two data sets are chosen because of the following reasons: (1) The two data sets are from two kinds of typical threaded discussion communities. One is the Q&A forum, and the other is an open discussion forum where everyone can participate and judge the comments. Both of them have time stamps in each post, and the reply structure can be extracted without much difficulty. (2) These two data sets are all or at least partial labeled. The iPad Q&A data set can label the answers "Helpful" by other users or "Solved" by the questioner, while *Slashdot.org* can give each comment a score ranging from -1 to 5 by all the participators. The quality vector $\mathbf{q}$ and the weight α in final fusion are acquired by supervised learning, which relies on the labeled data. For each data

Table 1. The basic statistics of the data sets

Data set	iPad Q&A	Slashdot.org
Number of threads	1130	664
Number of posts	8489	146569
Number of users	2175	14241
Average thread length	7.51	220.74
Average words per post	63.09	76.33
Average posts per user	3.90	10.29
Timestamp(mins from 1970)	21075992 - 21590652	22091472 - 22633163
Number of topics	5	5

sets, we select 5 hottest topics and ignore the unqualified threads that have posts fewer then 3 or without labels or ratings. The basic statistic results are shown in Table 1, from which we know that the two kinds of threaded discussion communities are quite different in average thread length, users active degree and so on. However, proving the similarity in content and structure organization, we can get the valuable answers to the questions or recommend the popular comments in our unified fusion framework.

6.1 Evaluation for the Time-Related Rank Model

Our model is different from the previous studies largely because we consider the time stamp which represents the timeliness of the post in a thread. There are two time intervals considered in our trr model. One is how long the post has stayed on the webpage until now, the other is between the post and its reply posts. Let us take iPad Q&A data set as an example. As shown in Fig. 4, every post belongs to one thread and has a time stamp that represents the released time. Figure 4a shows that the distribution of the posts in each thread in the time line. Every post is either unlabeled or labeled with "Helpful" or "Solved". Figure 4b shows the time intervals between post and its reply posts follow the power law distribution. The most of the time intervals between reply posts are less than a few hours. When the time interval becomes large, the number quickly decreases.

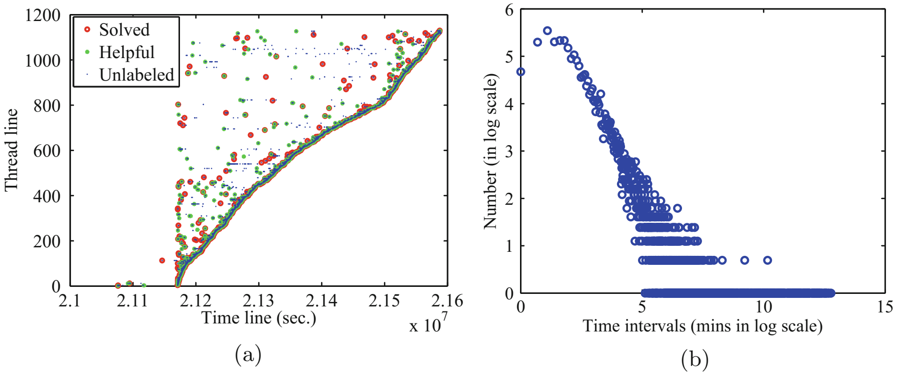


Fig. 4. Timeliness in iPad Q&A data set

In the experiment with the trr model, we define the time interval function $K(ts_u, ts_v) = \log(ts_u - ts_v)$ between post v and its reply post u. The timeliness of post v is $h(v) = H(ts_v)$, which can be calculated as

$$H(ts_c, ts_v) = \exp\left(-\frac{ts_v - ts_{min}}{ts_{max} - ts_{min}}\right) \tag{8}$$

As for each thread, ts_{max} is the time stamp of the timeliest post and ts_{min} is the latest.

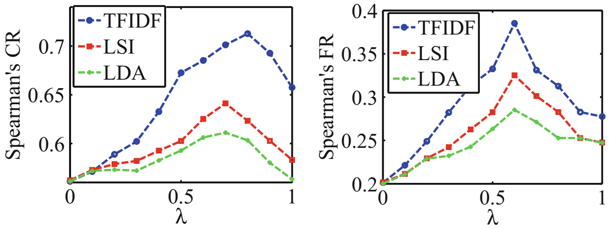
Table 2. The rank evaluation results

Data set	iPad Q&A				Slashdot.org			
Criteria	TS(ASC)	TS(DESC)	PR	TRR	TS(ASC)	TS(DESC)	PR	TRR
Spearman's RC	0.8449	0.5859	0.7856	0.8378	0.3247	0.1108	0.4648	0.5119
Spearman's FR	0.8418	0.5882	0.7824	0.8351	0.3212	0.1156	0.4664	0.5111

In the iPad Q&A data set, the post in a thread can only be one of three labels, "Unlabeled", "Helpful" or "Solved". We rank them 3, 2, 1 respectively. While in *Slashdot.org*, the posts those are marked a score from -1 to 5 are directly ranked from 7 to 1. We rank the posts in a thread according to four criteria:(1) time stamp in ascending order (2) time stamp in descending order (3) PageRank score (4) trr score, as shown in Table 2. Mallows model with Spearman's rank correlation and footrule [12] is introduced to measure the results. The iPad Q&A data set is from a forum that many senior iPad users or even service staff answer the questions timely. As a result, when the question is released by a green hand, it can be solved the first time, which is shown in the table that rank the time stamp in ascending order performs best. Our trr model performs much better then PageRank because we take the timely release and timely reply into consideration. While on the dataset from Slashdot.org, our trr model performs best.

6.2 The Semantic Reconstruction Experiments

We select three models TFIDF, LSI and LDA to conduct the semantic reconstruction. In the similarity calculation (see Eq. 2), the parameter λ balances the angle and the length of two post vectors. We carry out an experiment to choose the best λ for three models on each data set. As shown in Fig. 5, we choose λ for the three models $\lambda_{iPad} = (0.8, 0.7, 0.7)$ on iPad Q&A data set, and $\lambda_{Slashdot} = (0.6, 0.6, 0.6)$ on *Slashdot.org*.



(a) λ for iPad Q&A

(b) λ for Slashdot.org

Fig. 5. Find the λ for each data set

6.3 Evaluation for the Unified Fusion Framework

The quality measure follows the training paradigm. For each data set, we randomly choose $\gamma = 0.2$ of the whole threads to train. The similarity parameters on each data set in the semantic reconstruction are adopted in our quality measure. The topic numbers that we choose in LSI and LDA are both 5. As shown in Table 3, the quality measure of three semantic models on two strategies is consistent, the quality of TFIDF is the best because the posts are short in threaded discussion communities. With words in a post as much as possible, we can get the semantics of the post much better. While the disadvantage is also obviously, TFIDF is more time-consuming then other two models.

Table 3. The quality measure for the unfied fusion framework

Data set	iPad Q&A			Slashdot.org		
Quality	q_{tfidf}	q_{lsi}	q_{lda}	q_{tfidf}	q_{lsi}	q_{lda}
Combine then rank	0.51	0.41	0.08	0.66	0.25	0.09
Rank then combine	0.54	0.38	0.08	0.69	0.22	0.08

The training process also gets the fusion parameter α between semantics and structure incorporating quality measures. Based on all the above parameters we get from the experiments, we get the final fusion time-related rank result on the remaining test data set. As shown in Table 4, our fusion framework which combines both semantics and structure information of the thread performs much better than the model just from structure or semantics in time-related rank.

Table 4. The rank results for the unified fusion framework

Data set	iPad Q&A				Slashdot.org			
Criteria	St	Se(CR)	Se(RC)	Fusion	St	Se(CR)	Se(RC)	Fusion
Spearman's RC	0.8133	0.6228	0.6452	0.8456	0.4121	0.3373	0.3487	0.5521
Spearman's FR	0.8025	0.6182	0.6438	0.8424	0.3351	0.3256	0.3412	0.5414

7 Conclusions

We have described a time-related rank model in our paper, which takes the time stamp of the post into consideration. Based on the assumption that the post which is timely released and with large posts replying immediately should rank high, we have designed an algorithm to alleviate the influence of the time factor when rank different posts with different time stamps. We have also proposed a method to reconstruct the structure of the posts in a thread according to their semantics. Finally we have constructed a unified framework to fuse different models incorporating quality measures. In the unified fusion framework, the structure and semantics of a thread with many posts can be easily adopted. Experiments on two data sets from two kinds of typical threaded discussion communities have demonstrated that our time-related rank model works better than PageRank. The unified fusion framework is also easily extended when new features or models are added.

Acknowledgments. This work is partly supported by the 973 basic research program of China (Grant No. 2014CB349303), the National 863 High-Tech R&D Program of China (Grant No. 2012AA012504), the National Science Foundation of China (Grant No. 61379098, No. 61103056), the Natural Science Foundation of Beijing (Grant No. 4121003), and The Project Supported by Guangdong Natural Science Foundation (Grant No. S2012020011081).

References

1. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. JASIS **41**(6), 391–407 (1990)
2. Gee, K.R.: Using latent semantic indexing to filter spam. In: Proceedings of the 2003 ACM Symposium on Applied Computing, pp. 460–464. ACM (2003)
3. Baron, J.R.: Law in the age of exabytes: Some further thoughts on ‘information inflation’ and current issues in e-discovery search. Rich. JL Tech. **17**, 9–16 (2011)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
5. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web (1999)
6. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM (JACM) **46**(5), 604–632 (1999)
7. Cong, G., Wang, L., Lin, C.Y., Song, Y.I., Sun, Y.: Finding question-answer pairs from online forums. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 467–474. ACM (2008)
8. Blei, D.M., Moreno, P.J.: Topic segmentation with an aspect hidden markov model. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 343–348. ACM (2001)
9. Shen, D., Yang, Q., Sun, J.T., Chen, Z.: Thread detection in dynamic text message streams. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 35–42. ACM (2006)
10. Xu, G., Ma, W.Y.: Building implicit links from content for forum search. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 300–307. ACM (2006)
11. Lin, C., Yang, J.M., Cai, R., Wang, X.J., Wang, W.: Simultaneously modeling semantics and structure of threaded discussions: a sparse coding approach and its applications. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 131–138. ACM (2009)
12. Spearman, C.: The proof and measurement of association between two things. The Am. J. Psychol. **15**(1), 72–101 (1904)

Multi-group-based User Perceptions for Friend Recommendation in Social Networks

Trung L.T. Nguyen[1] and Tru H. Cao[1,2(✉)]

[1] Ho Chi Minh City University of Technology – VNUHCM,
Ho Chi Minh City, Vietnam

[2] John von Neumann Institute – VNUHCM, Ho Chi Minh City, Vietnam
tru@cse.hcmut.edu.vn

Abstract. Friend recommendation is one of the most important services of social networks. There is a great interest in determining a user's perception, which is a set of social features based on which the user make friends with others, to provide high quality recommendations. This perception varies from person to person. With various types of relationship, it would also be changed. In this paper, we present a method to recommend friends in social networks based on user's perception in each of his/her friend groups. We use social genomes to represent friend group perceptions. We conduct experiments on a Facebook dataset obtained by using our Facebook application and Facebook Graph API. The results show that the proposed method using perception for each group outperforms the prior one that use only one perception for all friends of a user.

1 Introduction

Social networks have become more and more popular. Well-known social networking websites on the Internet can be listed as Facebook, Twitter and Google+. To some extent, they are also contributing to a change in human social behaviors. In social networks, friend recommendation systems play an important role in providing quality customized user experiences. They are changing the way people interact with the Web by providing more personalized information access experiences than searching [14].

The main challenge in developing a relevant friend recommendation system is the dynamic nature of human perception of friendship [2, 5]. That perception varies from person to person [13]. Another significant point is that, with various types of relationship, it would also be changed. That means, in two different groups, the perceptions of friendship of a user may be also different. The goal of our research here is to gain insights into the preferences that a user considers when forming relationship with friend groups by studying human interaction within social networks, in order to provide better quality, i.e., more relevant, friend recommendations.

In the literature, the network-based approach is mostly used and generally performs well in providing high quality recommendations [4]. Friends-of-friends is a method in this approach, which implies that the probability that a person gets a new friend via their friends is higher than directly with an arbitrary person [7]. That is, if a person is a friend of a friend of a user, then that person is a good candidate friend to be

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 525–534, 2014.
DOI: 10.1007/978-3-319-13186-3_47

recommended to that user. It appears that popular social networks such as Facebook, Twitter, or LinkedIn employ friends-of-friends for friend recommendation. In [9], the authors proposed a method that used a genetic algorithm to optimize three indices derived from structural properties of social networks in solving the friend recommendation problem. In [8], the authors described a friend suggestion algorithm that used a user's implicit social graph. That implicit social graph was built based on virtue interaction of a user with his/her friends. However, those methods do not provide any insight into human cognitive components [4].

Another approach is social-based. It focuses on estimating a particular user's interests from the data that are implicitly or explicitly generated by the user. In [15], the authors presented a friend recommendation system using a preference model based on user rating of several portrait photos. In [14], the authors measured the similarity between individuals in terms of their location histories and recommended a group of potential friends to a user. However, the social-based approach is not as efficient as the network-based one [4].

Meanwhile [4] presented a method that combined network topology and a genetic algorithm. In that work, to recommend friends to a user, the friends-of-friends method was used as a filtering step to remove irrelevant individuals. A social genome was used to represent the friendship perception of the user. That is, such a social genome was used to examine the user's possibility to pursue a relationship with an individual. The work's result showed that combination of network-based and social-based approaches was more effective in recommendation, in comparison to each of its individual counterparts. Besides, its contribution was improvement of friend recommendation based on user perception.

In [4] only one perception was used for all current friends of a target user to whom new friends are to be recommended. However, in reality, a user usually has different friend groups with different perceptions. Moreover, he/she actually gets new friends via each of his/her friend group that shares some common interests. For example, for the "neighbor" group of a user, a common interest could be "nearby living place'. Meanwhile, in a "sport association" group, a common interest could be "good health" that sports may bring back.

Therefore, in this paper, we propose a method that employs user perceptions in separate groups to recommend friends in social networks. That means it looks for the features in each friend group that attract a user in creating relationship with new friends via that group. It is expected that obtained multi-group-based perceptions would be more relevant to friend recommendation to a user than a single perception of all friends of that user.

The details of our method are explained in Sect. 2. Experimental results are presented and discussed in Sect. 3. Finally, in Sect. 4, we conclude the paper and discuss some directions for future work.

2 Proposed Method

As mentioned above, user perception of friendship is significant to friend recommendation, and it changes across different contexts, i.e., friend groups. Our proposed method departs from [4] with using multiple perceptions for different friend groups of a target user, instead of just a single perception for all current friends of that user. That is, the method first detects the user's friend groups and determines his/her perception for each of the detected groups. The new friends to be recommended to that user will be obtained via each of his/her friend group using the friends-of-friends method and the user perception of that group. In Sect. 2.1, we present our employed method to detect clusters of friends of a user. Section 2.2 defines social genomes to represent friend group perceptions. Section 2.3 presents the final filtering step to recommend friends to a user.

2.1 Friend Group Detection

One could consider a social network as a graph with communities in which the vertices correspond to the individuals in the network, and the edges correspond to the friend relation between individual pairs. Not as in a random graph, in a social network graph, edge distribution is not uniformed. That forms separate groups of vertices in each of which there is a high density of connecting edges, while there are much fewer edges connecting those groups [6]. In each group, the vertices probably share common properties and/or play similar roles within the graph [10]. Therefore, if an individual has similar properties to the common ones of a group, he/she is more likely to be connected to the group than an arbitrary individual.

In order to recommend appropriate new friends to a target user, our proposed method first explores current friend groups of that user. To detect friend groups of a user, we use the Markov clustering algorithm (MCL) that simulates the diffusion process of random flows in a graph, in which "a random walk that visits a dense cluster will likely not leave the cluster until many of its vertices have been visited." [11]. By doing many random walks on the graph, they will gather in and form groups of vertices.

As shown in [12], MCL produces good results in most of cases, but its run time increases significantly when a graph size goes large. However, for our method, it is used only for small friend groups of a certain user. According to Facebook statistics, the average number of friends of a Facebook account is only 130. Also, due to the simplicity of this algorithm, we employ it for detecting user's friend groups. The next step is to determine the common features of each friend group of a user as a basis to recommend new friends to that user.

2.2 User Perception Discovery

Commonly, a person likely pursues a relationship based on similar features or similar preferences (e.g., similar musical tastes, similar hobbies, etc.) [13]. That means, in perception of a person, he/she tends to get relations with those who have similar features or similar interests. It is natural and frequent for humans to change their perception of friendship. This perception varies from person to person. In relations with different friend groups, this perception is also changed. For examples, in a social network, when setting relationship with friends in "neighbor" group, a user usually pays attention on "nearby living place" of his/her friends. But, in "sport association" group, a user tends to communicate with someone who has similar "interest" in sport. Or, in "schoolmate" group, “age range” and “education” are concerned features. As shown in our experiments presented later in this paper, an individual whose features are close to the common ones of a particular friend group of a user has a higher probability to become that user’s new friend than another individual whose features are close to the common ones of all friends of that user.

In this paper, user perception of friendship is represented by a binary genome whose genes are based on certain social features. The social features are preferences that a user may apply in the decision to pursue a friendship. As in [4], each bit of a genome corresponds to a feature. Its value is set to 1 or 0 depending on whether the corresponding feature is the same as that of the user or not.

In our research on Facebook, the social features that we consider are the followings obtained using our developed Facebook application and Facebook Graph API:

1. *Shared Friends:* the more friends an individual shares with a user, the more likely that individual becomes a new friend of the user. If an individual shares at least one friend with the user, the corresponding bit of the individual’s genome is set to 1; otherwise, it is set to 0.
2. *Location:* living nearby each other helps making friends. If the location of an individual is the same as that of the user, then the corresponding bit of the individual’s genome is set to 1; otherwise, it is set to 0.
3. *Age Range:* individual ages are divided into ranges such as [15, 20], [21, 25], and so on. If an individual and the user have the same age range, then the corresponding bit of the individual’s genome is set to 1; otherwise, it is set to 0.
4. *Gender:* people of the same gender usually have similar interests and thus tend to make friends with each other. If an individual has the same gender as the user, then the corresponding bit of the individual’s genome is set to 1; otherwise, it is set to 0.
5. *General Interests:* one tends to make friends with another having common general interests. If an individual shares at least one general interest with the user, the corresponding bit of the individual’s genome is set to 1; otherwise, it is set to 0.
6. *Education:* people who once studied together in the same school will likely become friends later. If an individual and the user were once in the same school, then the corresponding bit of the individual’s genome is set to 1; otherwise, it is set to 0.
7. *Work:* like education, working in the same place is also a good condition for making friends. If an individual works in the same organization as the user, then the corresponding bit of the individual’s genome is set to 1; otherwise, it is set to 0.
8. *General Groups:* Facebook allows a user to create or join in a group through which the user may gets new friends. If an individual is in at least one common group with the user, then the corresponding bit of the individual’s genome is set to 1; otherwise, it is set to 0.

Figure 1 illustrates an example social genome where active genes, whose values are 1, include *Shared Friends*, *Location*, *Gender*, and *General Groups*. Actually, there are other features that may affect friendship creation [1, 5]. In [4], the authors used 10 features, among which there are 4 features that are not in our list of 8 features described above, namely *Photo Tags*, *Events*, *Movies*, and *Religion/Politics*. However, in [4] the used features were only to demonstrate that using user perception of friendship improves friend recommendation, while our work here is to show that using multi-group perceptions gives better recommendation results than using a single perception. Moreover, not every social feature has its personal information available on a social network like Facebook to be used for experiments.

1	2	3	4	5	6	7	8
1	1	0	1	0	0	0	1

Fig. 1. An example of social genomes

Given a friend group of a user, in order to discover the user's perception about individuals in the group, we employ the Niched Pareto Genetic Algorithm (NPGA) [3], which is an advancement of genetic algorithms for multi-objective problems, i.e., having more than one fitness function. For the friend recommendation problem here, there are two objective functions. That is, the solution is the group representative genome that has the most number of active genes and the most number of individual genomes satisfying it. An individual genome is said to satisfy the group representative genome if and only if the active genes of the former include those of the latter.

2.3 Friend Recommendation

For a certain user, after detection of the user's friend groups and discovery of the user's perception in each group, new friends could be recommended for the user as follows. First, candidate friends via a detected friend group are those obtained by using the friends-of-friends method. Second, for each friend group, a candidate whose genome matches with the group representative genome will be shortlisted. Finally, the top individuals in each shortlist are recommended friends to the user.

Figure 2 presents the pseudo-code of our proposed friend recommendation method based on user group perceptions. The input includes a target user u to whom new friends are recommended, the set F of current friends of u, and the set P of individuals who are not u's friends yet. The output is recommended friends to u, assuming that the maximum number of recommended friends is 10. In line 1, MCL algorithm is used to detect friend groups of u, as presented in Sect. 2.1. In line 4, the representative genome for each group is discovered using NPGA. In line 5, for each detected group, the friends-of-friends method is used to select candidate friends of the user. In lines 6–11, candidate friends are scored and ranked using the user perception-based method in [4], and the top ones are shortlisted. Finally, in lines 13–15, the best ranked individuals from each shortlist are recommended, so that the total number of recommended friends is 10.

3 Evaluation

In the previous sections, we introduce our proposed method in friend recommendation by using more group representative genomes than in [4] to express multi-perceptions of friendship with different friend groups of a target user, to whom new friends are to be

```
Input: User $u$ to whom new friends are recommended, $F$ is the
set of all $u$'s current friends, $P$ is the set of individuals
who are not friends of $u$ yet.
Output: 10 recommended friends to user $u$.
$G \leftarrow$ MCL($F$)
$I \leftarrow$ null
for each group $g$ in $G$
    Finding the representative social genome of $g$ using
    NPGA
    $P_g \leftarrow$ Friends-of-friends($P$, $g$)
    for each individual in $P_g$ do
        Scoring individual based on his/her genome
        and the group representative genome
    end for
    Sorting $P_g$ according individual scores in the de-
    scending order
    $I_g \leftarrow$ pop top 10 of $P_g$
    $I = I \cup \{I_g\}$
end for
recommended friends $\leftarrow$ null
while count(recommended friends)<10
    recommended friends += pop top of each $I_g$ in $I$
```

Fig. 2. Friend recommendation using multi-group-based user perceptions

recommended. We now present the evaluation of the proposed method against the prior one that uses only one representative genome for all current friends of a user.

3.1 The Dataset

For our experiments, a Facebook user dataset consisting of 13,403 nodes and 225,223 edges is built by using our developed Facebook application and Facebook Graph API. Target users and the population from which friends are selected for recommendation to those users are covered by this dataset. The goal of our experimentation is to show that if different social genomes are used to represent user perceptions in different friend groups, then better friend recommendation can be achieved.

3.2 Experiment Setup

In [4], 100 target users from a Facebook sub-network consisting of 1,200 users were chosen for evaluating the friend recommendation method therein. For each of those 100 users, 10 of his/her current friends were randomly removed and that method would then recommend 10 replacing ones for that user. Recommended friends are said to be relevant to a target user if they match with the 10 removed ones for that user in the test dataset. Therefore, the precision of a method is calculated by the following formula:

$$Precision = \frac{Number\ of\ revelant\ recommended\ friends}{Total\ number\ of\ recommended\ friends}$$

That is, ideally, the precision of a method would be 100 % if all removed friends are recommended by the method. Since the number of removed friends and the number of recommended friends are equal, which is 10 in this case, the Recall measure is the same as the Precision one.

Similarly, we choose 80 users in the built Facebook dataset presented above to recommend friends, also with 10 removed current friends for each of them and 10 recommended friends by our proposed method. Those 80 target users are chosen because we could obtain their complete current friend lists each of which has at least 50 friends (so that finding a representative genome makes sense), and their current friends have sufficient information on the 8 selected features for determining each group representative genome.

Due to the common characteristic of genetic algorithms, the discovered representative genome of each friend group may slightly vary across different runs, which also affects friend recommendation results. Therefore, in our experiments, for each target user, the recommendation precision of a method is taken as the average of those in different 5 runs.

3.3 Experiments

We have tested and compared the performance of the two following methods:

Method A. Recommending friends to a user using a single perception about all current friends of that user (as in [4]).

Method B. Recommending friends to a user using multi-perceptions about different friend groups of that user (our proposed method).

Table 1. Experimental results. Columns (*a*) and (*b*) respectively show the average precisions of method *A* and method *B* on 80 Facebook target users.

	Method *A (a)*	Method *B (b)*
Average Precision on 80 users	35.85 %	63.53 %

Table 1 presents the average precision on 5 runs of each method on 80 chosen Facebook target user accounts. It shows that the proposed multi-perception-based method outperforms the single-perception-based one.

Meanwhile, Fig. 3 illustrates the statistics of friend recommendation results using the two methods. The horizontal axis represents the number of relevant recommended friends. The vertical axis represents the number of target users having those numbers of relevant recommendations. For example, as shown in the chart, using method *B* there are 19 users having 6–7 relevant recommendations, while there are only 7–9 users if using method *A*.

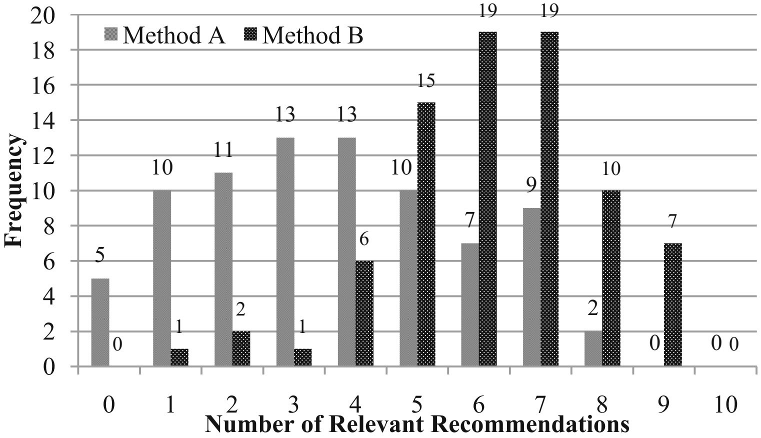


Fig. 3. Histograms showing the frequencies of users who have a certain number of relevant recommendations by method *A* and method *B*

Figure 4 shows the recommendation precision of each method on each target user. The users are arranged on the horizontal axis according to their recommendation precision using method *B* in the ascending order. It shows that, for every user, the precision using method *B* is higher than that of method *A*.

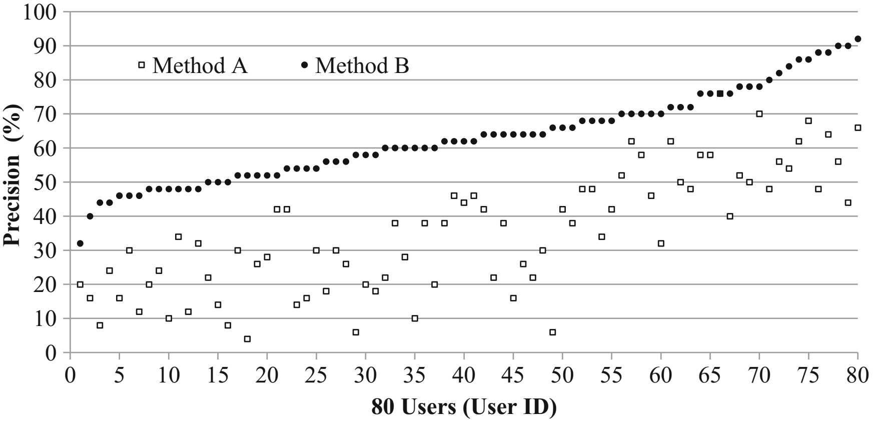


Fig. 4. Recommendation precisions for all users arranged in the ascending order with respect to method *B* along with the corresponding precisions of method *A*

4 Conclusion

In this paper, we have proposed to use multi-group perceptions for friend recommendations in a social network, instead of using a single perception in previous work. Our proposed method first detects the current friend groups of a user and discovers the user perception for each group, which is then used to recommend new friends to that user. The experiment results have shown that our method outperforms the prior one using only a single perception about all of the current friends of a user. Furthermore, our research has provided an insight, in terms of individual features, as to how and why each particular user may form relationship with a certain group of friends.

However, the current approach may not work as expected in certain cases. One case is that, in a social network like Facebook, users have the option of excluding information from their profiles and, furthermore, may post false information. Another case is the cold start problem of new comers in a network, who have just a few friends. In such cases, there may not be sufficient truthful information to determine user perception to be used for friend recommendation. Dealing with little or uncertain user profiles is among the topics that we are working on.

References

1. Jung, C.G.: Analytical Psychology: It's Theory and Practice. Vintage, New York (1970)
2. Nowell, D.L., Kleinberg, J.: The link prediction problem for social networks. J. Am. Soc. Inform. Sci. Technol. **58**(7), 1019–1031 (2007)
3. Horn, J., Nafpliotis, N., Goldberg, D.E.: A niched pareto genetic algorithm for multiobjective optimization. In: Proceedings of the 1994 IEEE World Congress on Computational Intelligence, vol. 1, pp. 82–87 (1994)
4. Naruchitparames, J., Güneş, M.H., Louis, S.J.: Friend recommendations in social networks using genetic algorithms and network topology. In: Proceedings of the 2011 IEEE Congress on Evolutionary Computation, pp. 2207–2214 (2011)
5. Argyle, M.: Social Interaction. Library of Congress, Washington (2009)
6. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. U.S.A. **99**(12), 7821–7826 (2002)
7. Mitchell, M.: Complex systems: network thinking. Artif. Intell. **170**(18), 1194–1212 (2006)
8. Roth, M., et al.: Suggesting friends ussing the implicit social graph. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 233–242 (2010)
9. Silva, N.B., Tsang, I.R., Cavalcanti, G.D.C., Tsang, I.-J.: A graph-based friend recommendation system using genetic algorithm. In: Proceedings of the 2010 IEEE Congress on Evolutionary Computation, pp. 1–7 (2010)
10. Boccaletti, S., et al.: Complex networks: structure and dynamics. Phys. Rep. **424**, 175–308 (2006)
11. Dongen, S.V.: Performance criteria for graph clustering and Markov cluster experiments. Technical report INS-R0012, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam (2000)
12. Brandes, U., Gaertler, M., Wagner, D.: Experiments on graph clustering algorithms. In: Di Battista, G., Zwick, U. (eds.) ESA 2003. LNCS, vol. 2832, pp. 568–579. Springer, Heidelberg (2003)
13. Liyong, W.: An adaptive evolution mechanism for growing social networks. In: Proceedings of the 2008 International Conference on Information Management, Innovation Management and Industrial Engineering, vol. 2, pp. 320–324 (2008)

14. Zheng, Y., et al.: Recommending friends and locations based on individual location history. ACM Trans. Web **5**, 1–44 (2011)
15. Wu, Z., Jiang, S., Huang, Q.: Friend recommendation according to appearances on photos. In: Proceedings of the 17th ACM International Conference on Multimedia, pp. 987–988 (2009)

STC: A Joint Sentiment-Topic Model for Community Identification

Baoguo Yang and Suresh Manandhar(✉)

Department of Computer Science, University of York, York, UK
by550@york.ac.uk, suresh@cs.york.ac.uk

Abstract. Traditional methods for identifying communities in networks are based on direct link structures, which ignore the content information shared among groups of entities. Recently, community detection approaches by using both link and content have been studied. It is necessary to identify communities with different sentiment distributions based on corresponding topics, which cannot be identified by existing community discovery techniques. To directly detect the sentiment-topic level communities and to better explore the hidden knowledge within them, we propose to integrate social links, content/topics, and sentiment information to work out a novel community model. Experimental results on two types of real-world datasets demonstrate that our model can not only achieve comparable performance compared with a state-of-the-art community model, but also can identify communities with different topic-sentiment distributions.

1 Introduction

The rapid growth of social medias provide us more chance to contact with other people and share our interests and opinions online, such as Facebook, Myspace, Twitter, etc. Email is considered as another kind of communication tool, which brings us more convenience to send or receive messages. A huge amount of data are generated online every day. Discovering previously unknown knowledge and relationships among people is very useful and necessary for individuals and organizations.

Example 1 (Email Networks): Email is widely used in our daily life, especially in companies and universities. Email correspondence produces abundant social messages associated with social relations. For teachers, their email recipients can be students, colleagues, friends, family members, librarians, and book publishers, etc. To get a high-level overview of the emails in our mailboxes, it is very interesting and necessary to discover our social communities in an automatic way. In each community, we are interested in the topics we discussed, people we contacted with, and the sentiment on some topics. Such information is latent and unobservable.

Example 2 (Hotel Twitters): Twitter, a popular microblogging platform, is not only used by individuals, but also very popular in many organizations,

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 535–548, 2014.
DOI: 10.1007/978-3-319-13186-3_48

such as companies, hotels, and online supermarkets. As we know many hotels have their own twitter accounts. The customers can send their tweets about opinions and reviews to the hotels, and can comment on other tweets about the environment, food, and service of the hotels. To make full use of the data, it is useful to automatically identify communities associated with this twitter account. The communities with obvious negative polarities should be considered firstly. The hotel managers can take actions to address the main issues these customers proposed, and then response to these groups of people about the quality improvement of the hotel to win more customers, and to avoid the negative information proliferation across communities. Note that if we only extract collections of tweets including same sentiment topics by using traditional sentiment analysis methods instead of mining communities, the important social links will be ignored.

Based on the above examples, it is demanding to devise an effective community discovery approach to tackle these issues. The research on communities has a long history, and it has been paid widely attention in the past decade. In [2,9], Girvan and Newman propose a popular divisive community detection algorithm based on the concept of betweenness. To improve the speed of the algorithm in [2], a modified algorithm is proposed by Tyler et al. in [15]. Also some overlapping community detection methods has been proposed, like [4,17]. In addition, dynamic community discovery has been studied in recent years [3,10], where communities are not static but evolve over time.

However, most of the existing community identification methods intend to learn the community structures just using links, which ignore the content information in social networks. In recent years, the research on community detection has attracted increasing attention and achieved great progress. Discovering communities by combining link and content has been proposed in the literature [12,14,18–20], however, these methods fail to consider the valuable sentiment information in social networks.

In this paper, we propose a novel ***S****entiment-*_**T**_*opic model for* ***C****ommunity discovery*, called STC, which is built by using social links, topics and sentiment in a unified way, where the sentiment is studied based on its corresponding topic. The main goal of this approach is to discover sentiment level communities, i.e., to find out some communities containing dominant sentiments on certain topics even though not all communities have dominant sentiment topics. In our model, we define a community as a collection of people who are directly or indirectly connected and share some sentiment topics with some members in this collection. Note that not all the topics are discussed by every member of the community, also not all the members have the identical sentiment towards a certain topic, and the connectivity among members is also a very important factor. In many cases, even if two groups of people have similar sentiment-topic distributions, they are not included in the same community when the two groups follow different user distributions.

The rest of this paper is organized as follows: Sect. 2 introduces the related work. We present our community discovery model, the generative process and parameter estimation in Sect. 3. In Sect. 4, we present and discuss the experimental results on two real-world datasets, the comparison with an up-to-date model is also reported. We give short discussion in Sect. 5, and the conclusions with future work are presented in Sect. 6.

2 Related Work

Traditional algorithms are focused on identifying disjoint communities [2,9], while in many real-world networks communities are allowed to overlap to some degree, where an entity can be included in multiple communities. The clique percolation method proposed by Palla et al. [11] is an early technique for overlapping community detection. Later, many algorithms have been proposed to improve the performance of the detection methods, such as OSLOM [4], SLPA [17], etc.

The above mentioned community identification methods ignore the content of social interactions in social networks. An early framework for community discovery using link and content elements is proposed in [19], the authors proposed two community-user-topic (CUT) models based on joint user and topic distributions. In [18], Yang et al. propose to integrate a popularity-based conditional link model with a discriminative content model into a unified framework to discover communities. For maximum likelihood inference, a novel two-stage optimization algorithm is proposed.

CART (Community-Author-Recipient-Topic) [12], a Bayesian generative model, is proposed to integrate link and content information in the social network for discovering communities, which is an extension of the Author-Recipient-Topic (ART) model [7]. It is assumed that the authors and recipients are generated from a latent group. Another novel method for detecting communities in social networks using links and content is proposed in [14]. In such method, the discussed topics, social links, and interaction types are all used to build several generative community models, namely, TUCM (Topic User Community Model), TURCM-1 and TURCM-2 (Topic User Recipient Community Models) and full TURCM model. More recently, a community profiling model, Collaborator Community Profiling (COCOMP), has been proposed by Zhou et al. in [20] to identify the communities of each user and their relevant topics and groups. In COCOMP, both the social links and topics between users are also considered. In [8,13], content and links are also learnt together to identify communities.

However, the above methods fail to consider the sentiment information of topics, which is an important factor when discovering more meaningful communities on a level of sentiment. The joint sentiment/topic model (JST) [6], an extension of the traditional Latent Dirichlet Allocation (LDA) model [1], is proposed to detect document-level sentiment and topic from documents. In [5], Li et al. introduce two probabilistic joint topic and sentiment models, namely, Sentiment-LDA and Dependency-Sentiment-LDA. Sentiments are related to topics in both of the models. However, JST, Sentiment-LDA, and Dependency-Sentiment-LDA are not proposed for community discovery.

To overcome the above problems and identify more meaningful communities, we propose our community model, STC, using topic, sentiment and user interactions in a unified way, which takes the topic-sentiment into consideration.

3 Our Community Discovery Model

The graphical representation of our proposed community model, STC, is shown in Fig. 1. There are mainly two different variables in this model, the latent variables and the observable ones:

- The latent (hidden) variables: Community assignment c ($c = 1, 2, \cdots, M$); Topic assignment z ($z = 1, 2, \cdots, K$); Sentiment label assignment l ($l = 1, 2, \cdots, S$).
- The observable variables: Word w (the word in the document); Person u (the person who is sharing the document).

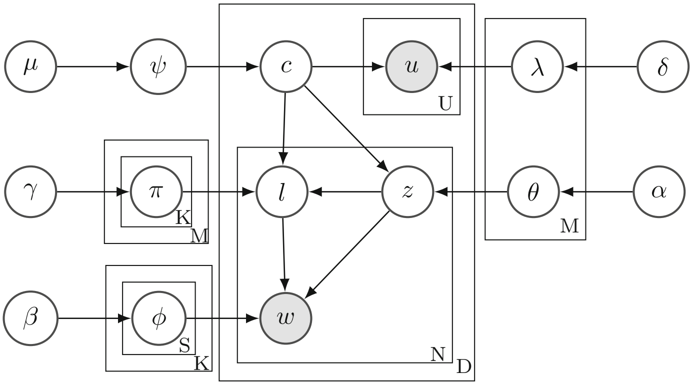


Fig. 1. Graphical notation of our proposed model.

3.1 Generative Process

Suppose there are K latent topics and S sentiment polarities, for each topic, and for each sentiment, we have: $\phi_{k,s}|\beta \sim Dir(\beta)$, where ϕ is the topic-sentiment distribution over words.

Let M be the number of communities, each community is related to three key parameters: (1) user participant mixture λ; (2) topic mixture θ; (3) sentiment mixture π. Specifically, in each community m ($m = 1, 2, ..., M$), θ_m is the topic mixture (proportion) for the community m, which follows a Dirichlet distribution $Dir(\alpha)$, λ_m is the user participant mixture with respect to community m, which has a Dirichlet distribution with hyperparameter δ. And $\pi_{m,k}$ is the sentiment mixture for topic k of community m. Note that the sentiments are studied based on topics, it is not reasonable to study sentiments without considering the corresponding topics. For example, given two topics “laptop” and “weather”, the sentiment words “nice” and “bad” can be used to describe both topics. It is not clear which topic is discussed by people with a sentiment word “nice” if the topic is not provided.

$$\theta_m|\alpha \sim Dir(\alpha), \quad \lambda_m|\delta \sim Dir(\delta), \quad \pi_{m,k}|\gamma \sim Dir(\gamma).$$

We define a community proportion ψ based on the whole corpus, $\psi|\mu \sim Dir(\mu)$. In this model, α, β, δ, γ, μ are the hyperparameters of Dirichlet distributions.

Then the generative process for each document d, $d = 1, 2, ..., D$ is shown as follows: Choose a community assignment c_d for a document d: $c_d|\psi \sim Mult(\psi)$.

Assume there are U_d people sharing a document d. For each person $u_{d,p}$ ($p = 1, 2, ..., U_d$) associated with document d, the generative process is: Choose a user $u_{d,p}$ from the participant mixture of community c_d: $u_{d,p}|\lambda, c_d \sim Mult(\lambda_{c_d})$.

Suppose there are N_d word tokens in a document d, For each word token $w_{d,n}$ ($n = 1, 2, ..., N_d$) in document d. The generative process is:

(1) Choose a topic assignment $z_{d,n}$ from the topic mixture of community c_d:

$$z_{d,n}|\theta, c_d \sim Mult(\theta_{c_d}).$$

(2) Choose a sentiment label $l_{d,n}$ from the c_d-th community’s sentiment mixture:

$$l_{d,n}|c_d, z_{d,n}, \pi \sim Mult(\pi_{c_d, z_{d,n}}).$$

(3) Choose a word $w_{d,n}$ from the distribution $\phi_{k,s}$ over words defined by the topic $z_{d,n}$ and sentiment label $l_{d,n}$: $w_{d,n}|z_{d,n}, l_{d,n}, \phi \sim Mult(\phi_{z_{d,n}, l_{d,n}})$.

From the graphical representation shown in Fig. 1, the joint probability for the proposed model can be written as Eq. 1.

$$\begin{aligned} &P(\mathbf{u}, \mathbf{c}, \mathbf{z}, \mathbf{l}, \mathbf{w}, \lambda, \psi, \theta, \pi, \phi|\delta, \mu, \alpha, \gamma, \beta) \\ &= P(\mathbf{u}|\mathbf{c}, \lambda)P(\mathbf{c}|\psi)P(\mathbf{z}|\mathbf{c}, \theta)P(\mathbf{l}|\mathbf{c}, \mathbf{z}, \pi)P(\mathbf{w}|\mathbf{z}, \mathbf{l}, \phi) \\ &\quad P(\lambda|\delta)P(\psi|\mu)P(\theta|\alpha)P(\pi|\gamma)P(\phi|\beta). \end{aligned} \tag{1}$$

3.2 Model Inference and Parameter Estimation

In this model, a document belongs to a single community rather than multiple communities. Each document is shared by at least two people (i.e., an author and at least one recipient) to make sure there is at least one link associated with a document. Once the sender (or the author) of the document is known, the user links associated with this document will be displayed. For inference, the statistics and variables are described in Table 1.

Let $t = (d, n)$, the conditional posterior probability of c_d, z_t, and l_t can be written as follows.

Table 1. List of statistics and variables.

Statistic/Variable	Description
D_m	the number of documents assigned to community m
D	the total number of documents
$n_{m,k}$ $(n_{m,k}^{-d})$	the number of times word tokens in the documents of community m are assigned to topic k (excluding document d)
$n_{m,k,s}$ $(n_{m,k,s}^{-d})$	the number of times word tokens in the documents of community m are assigned to topic k and sentiment label s (excluding document d)
n_m (n_m^{-d})	the total number of words in the documents of community m (excluding those in document d)
$n_{k,s,v}$ $(n_{k,s,v}^{-t})$	the number of times a word v is assigned to topic k and sentiment label s (excluding the word in position t)
$n_{k,s}$ $(n_{k,s}^{-t})$	the number of times words are assigned to topic k with sentiment label s (excluding the word in position t)
$f_{d,k}$	the number of word tokens in document d associated with topic k
f_d	the total number of words in document d
$f_{d,k,s}$	the number of word tokens in document d associated with topic k and sentiment label s
$n_{c_d,k}^{-t}$	the number of times word tokens in community c_d are assigned to topic k excluding the word in position t
$n_{c_d,k,s}^{-t}$	the number of times word tokens in community c_d are assigned to topic k and sentiment label s excluding the word in position t
$n_{c_d}^{-t}$	the total number of words in the documents of community c_d excluding the word in position t
$g_{m,p}$ $(g_{m,p}^{-d})$	the number of times a person p is involved in the documents of community m (excluding document d)
g_m (g_m^{-d})	the number of times persons are involved in the documents of community m (excluding document d)
$e_{d,p}$	the number of times a person p is involved in the document d
e_d	the number of persons who are sharing the document d
$\mathbf{l}_{d_{(k)}}$	the sentiment set of topic k in document d
$\mathbf{z}_d$	the topic set of document d
$\mathbf{u}_d$	the person set of document d

$$\begin{aligned}&P(c_d = m|\mathbf{c}_{-d}, \mathbf{u}, \mathbf{z}, \mathbf{l}, \mathbf{w})\\ &\propto \frac{D_m^{-d} + \mu_m}{\sum_{j=1}^{M} \mu_j + D - 1} \times \frac{\prod_{k\in\mathbf{z}_d} \prod_{i=0}^{f_{d,k}-1} (\alpha_k + n_{m,k}^{-d} + i)}{\prod_{i=0}^{f_d-1} (\sum_{k=1}^{K} \alpha_k + n_{m,k}^{-d} + i)} \\ &\times \prod_{k\in\mathbf{z}_d} \frac{\prod_{s\in\mathbf{l}_{d_{(k)}}} \prod_{i=0}^{f_{d,k,s}-1} (\gamma_s + n_{m,k,s}^{-d} + i)}{\prod_{i=0}^{f_{d,k}-1} (\sum_{s=1}^{S} \gamma_s + n_{m,k,s}^{-d} + i)} \times \frac{\prod_{p\in\mathbf{u}_d} (\delta_p + g_{m,p}^{-d})}{\prod_{i=0}^{e_d-1} (\sum_{p=1}^{P} \delta_p + g_m^{-d} + i)}.\end{aligned} \tag{2}$$

When the community assignment c_d for document d is obtained, for simplicity, the posterior distribution of z_t and l_t can be derived as follows.

$$\begin{aligned} &P(z_t = k, l_t = s|\mathbf{w}, \mathbf{z}_{-t}, \mathbf{l}_{-t}, c_d) \\ &\propto \frac{n_{c_d,k}^{-t} + \alpha_k}{\sum_{k=1}^{K} n_{c_d,k}^{-t} + \alpha_k} \times \frac{n_{c_d,k,s}^{-t} + \gamma_s}{\sum_{s=1}^{S} n_{c_d,k,s}^{-t} + \gamma_s} \times \frac{n_{k,s,v}^{-t} + \beta_v}{\sum_{v=1}^{V} n_{k,s,v}^{-t} + \beta_v}. \end{aligned} \tag{3}$$

The updated parameters are represented as follows:

$$\psi_m = \frac{D_m + \mu_m}{\sum_{m=1}^{M} \mu_m + D}, \quad \lambda_{m,p} = \frac{g_{m,p} + \delta_p}{\sum_{p=1}^{P} g_{m,p} + \delta_p}, \quad \theta_{m,k} = \frac{n_{m,k} + \alpha_k}{\sum_{k=1}^{K} n_{m,k} + \alpha_k},$$

$$\pi_{m,k,s} = \frac{n_{m,k,s} + \gamma_s}{\sum_{s=1}^{S} n_{m,k,s} + \gamma_s}, \quad \varphi_{k,s,v} = \frac{n_{k,s,v} + \beta_v}{\sum_{v=1}^{V} n_{k,s,v} + \beta_v}.$$

4 Experiment and Result Analysis

4.1 Experiment Setup

In the experiments, **two** types of datasets, the email dataset and the twitter microblog dataset are used. For Enron dataset[1], we randomly select five user folders, one of them called '*arnold-j*' is used for the experiment of individual user's perspective (denoted as arnold-j), and the other four folders, namely, *ermis-f*, *shively-h*, *whalley-g* and *zipper-a* are used together as a whole dataset (denoted as EnronFourUsrs). We conduct series of preprocessing work for arnold-j and EnronFourUsrs[2], like the initial duplicated email removal and the basic text mining preprocessing (stopwords removal, stemming, etc.). The second type of dataset is a twitter corpus[3], which includes 5513 tweets, covering 4 main topics, namely, Apple, Google, Microsoft, and Twitter. We kept the tweets belonging to one of the three sentiments (i.e., positive, negative and neutral), then the empty tweets and the ones without recipients are all removed. Some screen names are extracted from the text of tweets as the recipients, we also preprocess it to make the final document format the same as the Enron datasets. As for the four main topics in original twitter dataset, in fact, each main topic can be divided into several subtopics. The final preprocessed datasets for our experiments are shown in Table 2.

As the work in [5,6], we also use the subjectivity lexicons as prior information for model learning. Specifically, we use MPQA[4] [16] as the sentiment prior knowledge.

In our model, the initial values of the symmetric hyperparameters are set as: $\alpha = 50/K$, $\beta = \delta = \gamma = \mu = 0.1$. The collapsed Gibbs sampling algorithms are

[1] http://www-2.cs.cmu.edu/~enron/

[2] Note that we will use Enron to represent EnronFourUsrs in the following sections.

[3] http://www.sananalytics.com/lab/twitter-sentiment/

[4] http://www.cs.pitt.edu/mpqa/

Table 2. Basic information for the final datasets in the experiments.

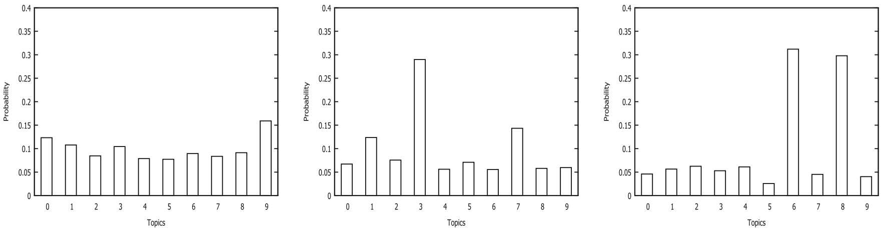
Dataset	# Docs	# Links	# Users
EnronFourUsrs	3804	38597	5623
arnold-j	2441	11474	2550
twitter	2247	3459	3460

executed 500 iterations to estimate the parameters in the models. The datasets are divided into two parts, 80 % of which are used for model training, and the rest are considered as held-out test set.

4.2 Analysis for Distributions Within Communities

In our model, each community has multiple topics, and each topic has multiple sentiment polarities, we studied the distributions within communities on different datasets.

Figure 2 gives the distribution of topics in individual communities. It can be seen from Fig. 2(a) that the topics are almost even within a single community 9 on Enron dataset. We also report selected communities on twitter dataset, in Fig. 2(b) and 2(c), some topics are dominant obviously in the communities. In Fig. 2(b), topic 3 (google android) is the dominant topic in community 1. In community 13, topic 6 (apple use) and topic 8 (iphone service) have large proportions, which are all the subtopics of "apple". These distributions imply that in some communities, people are only very interested in certain number of topics, which is in accordance with our main goal and community definition.



(a) Distribution of topics in community 9 of Enron dataset.

(b) Distribution of topics in community 1 of twitter dataset.

(c) Distribution of topics in community 13 of twitter dataset.

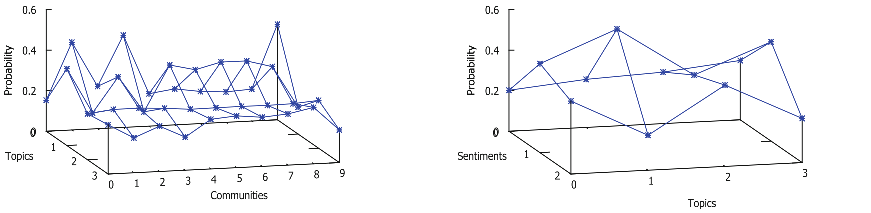
Fig. 2. Distribution of topics in individual communities, $M = 20$, $K = 10$.

Apart from the analysis on the topic distribution within selected individual communities, we also investigated the topic distributions for all the communities, and the sentiment distribution for all the topics in an individual community. Figure 3(a) and 3(b) give the topic and sentiment distributions on twitter

Table 3. Arnold-j's biggest community (community 4), $M = 5$, $K = 10$.

Topic ID	Topic	Positive	Negative	Neutral	people (denoted by the username of the enron email address)
4 (0.1337)	trading	0.3701	0.4498	0.1801	john.arnold (0.3746), jennifer.fraser(0.0282), ina.rangel(0.0217)
3 (0.1215)	power supply	0.5739	0.2403	0.1858	
5 (0.1167)	contract	0.3579	0.3363	0.3058	

dataset, respectively. It is obvious from Fig. 3(a) that different communities have nearly different topic distributions, although some topic distributions for some communities are a bit similar. As can be seen from Fig. 3(b) about the sentiment distribution for topics in community 0 that the sentiments for different topics can be different, which is common in real-world life that two communities may have different sentiment towards certain topics even if they have similar topic distributions (i.e., the two communities are talking about similar range of topics).



(a) Distribution of topics in all communities for twitter dataset.

(b) Distribution of sentiments of all topics in community 0 for twitter dataset.

Fig. 3. Distribution of topics within communities (sentiments for topics) for twitter dataset, $M = 10$, $K = 4$.

4.3 Community Analysis on Individual Users

We also studied the communities for a single user, arnold-j (*John Arnold*, a vice president in Enron company). Table 3 lists the largest community membership (community 4) for arnold-j, Column 1 and 2 show the main relevant topics and the corresponding probabilities within this community, columns 3–5 list the sentiment proportions for the corresponding topics, and the final column represents the top three active persons with high likelihoods in this community. It is obvious from Table 3 that the dominant sentiment polarity can vary with topics. Also we can see that *John Arnold* is the core people in this community.

In twitter dataset, we choose one entity with the screen name '@Apple' to study the hidden knowledge in its community. Table 4 shows the selected communities and sentiment topics that @Apple related to. Column 1 gives three selected participated communities, column 2 and 3 list the top two mainly discussed topics for each community with proportions, and the last three columns

Table 4. Selected communities of the user @Apple (ScreenName), $M = 20$, $K = 10$.

Community	Topic ID	Topic	Positive	Negative	Neutral
9	6 (0.3075)	iphone service	0.9152	0.0492	0.0356
	8 (0.2967)	apple use	0.9398	0.0335	0.0267
10	3 (0.2895)	google android	0.8445	0.0618	0.0937
	1 (0.1327)	twitter operation	0.6029	0.1972	0.1999
5	7 (0.1373)	microsoft	0.1595	0.7182	0.1223
	2 (0.1315)	twitter share	0.6311	0.2307	0.1382

describe the sentiment proportions for the corresponding topics. It is obvious from Table 4 that the mainly discussed topics among communities are different, which demonstrates that community 9, 10 and 5 are well identified, and also proves the effectiveness and feasibility of our model.

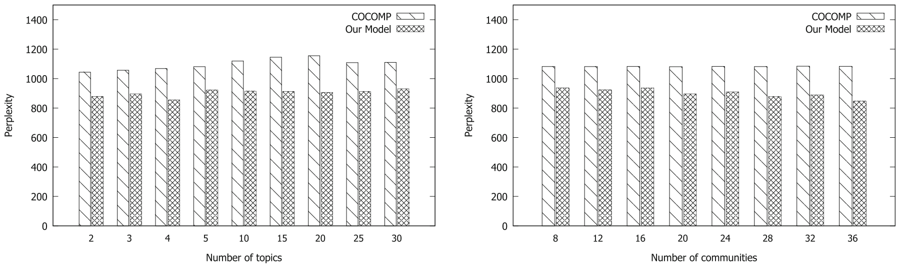
Based on the topics listed in Table 4, we show the top five words for each sentiment polarities of *topic* 1 and *topic* 6 in Table 5, each column lists a collection of highly ranked sentiment words and topic words. From these words, we can observe that topic 1 is about *twitter*, and topic 6 is about *apple*. It's a first attempt to detect sentiment-topic level communities via our STC model, while the sentiment information cannot be detected by the existing COCOMP model.

Table 5. Top ranked words for selected topics with different sentiments extracted by STC model.

Topic 1 (Twitter Operation)			Topic 6 (Apple Use)		
Positive	Negative	Neutral	Positive	Negative	Neutral
twitter	wrong	yeah	appl	account	touch
win	poor	custom	steve	site	babi
tech	troubl	absolut	job	close	player
world	mark	move	great	longer	feel
good	damag	launch	love	brand	report

4.4 Comparing with COCOMP Model

Note that the ground-truth communities are usually unavailable, which make the evaluation challenging. To evaluate our model, we also analysed the perplexity value, and made comparison with the state-of-the-art COCOMP model [20], which is a topic-level community discovery model. Each word in our model is determined by two factors, namely topic and sentiment, while there is only one factor, topic, for the COCOMP model. In our STC model, to generate a target word, both the topic and sentiment should be correctly assigned, otherwise the perplexity value will get worse, while only a correct topic assignment is required



(a) Perplexity under varying number of topics, $M = 20$.

(b) Perplexity under varying number of communities, $K = 5$.

Fig. 4. Perplexity results comparison between COCOMP and our model for twitter dataset.

in COCOMP model. The computation equations for the perplexity of our model is shown in Eq. 4. The lower perplexity tends to have the better performance.

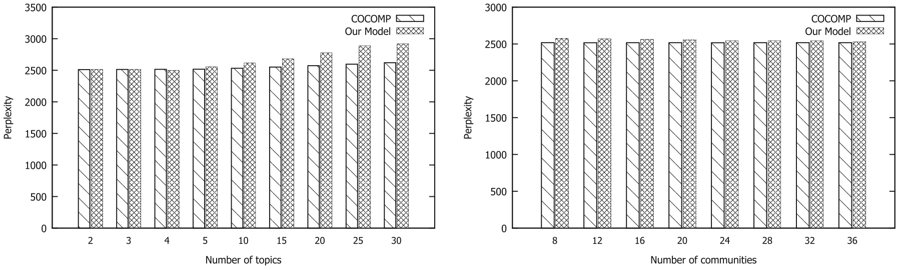
$$Perplexity(D_{test}) = \frac{\sum_{m=1}^{M} \log P(\tilde{\mathbf{w}}_m|\mathbf{w})}{\sum_{m=1}^{M} n_m}. \tag{4}$$

$$\begin{aligned} &P(\tilde{\mathbf{w}}_m|\mathbf{w}) \\ &= \prod_{n=1}^{n_m} \sum_{k=1}^{K} \sum_{s=1}^{S} P(w_n = t|z_n = k, l_n = s)\, P(l_n = s|z_n = k, c_{w_n} = m) P(z_n = k|c_{w_n} = m) \\ &= \prod_{t=1}^{V} \left(\sum_{k=1}^{K} \sum_{s=1}^{S} \phi_{k,s,t} \pi_{m,k,s} \theta_{m,k} \right)^{n_m^{(t)}}. \end{aligned} \tag{5}$$

$$\log P(\tilde{\mathbf{w}}_m|\mathbf{w}) = \sum_{t=1}^{V} n_m^{(t)} \log(\sum_{k=1}^{K} \sum_{s=1}^{S} \phi_{k,s,t} \pi_{m,k,s} \theta_{m,k}). \tag{6}$$

In Eq. 4, D_{test} shows the held-out testing documents, $\tilde{\mathbf{w}}_m$ denotes the words from testing documents appeared in community m, $\mathbf{w}$ represents the words in the training documents. n_m is the number of words in community m. As for Eq. 5, $n_m^{(t)}$ is the number of times a term t observed in community m, and c_{w_n} represents the community that the word w_n appears in.

The perplexity results for the two datasets are shown in Figs. 4 and 5. In each figure we illustrated the values of perplexity for our STC model and COCOMP with varying number of topics and communities. As can be seen from Fig. 4(a) and 4(b), the perplexity values of our model are lower than the COCOMP model. Although in Fig. 5(a) and 5(b), the perplexity value are worse than the COCOMP to some extent, it is still comparable to the COCOMP. Enron email and Twitter are two different types of social networking sites, the former is more



(a) Perplexity under varying number of topics, $M = 20$.

(b) Perplexity under varying number of communities, $K = 5$.

Fig. 5. Perplexity results comparison between COCOMP and our model for Enron dataset.

formal than the latter. Generally, there are more sentiment information in tweets than in emails. It is not the main concerning about which model has better perplexity value as long as our model has closer performance with COCOMP. Our model is proposed to identify sentiment level communities, which is not considered by COCOMP and other community discovery methods.

5 Discussions

We build our community discovery model, STC, by using social links, topics and sentiment information in a unified way. Those three factors are very significant to the identification of the meaningful community structures. However, it is not indicating that the more additional information incorporated into the model, the better result we can get. When the information is not important, the redundant factors can make the model more complex and inefficient. Not all the communities have sentiment information, our model is proposed to identify communities that have a certain degree of sentiment polarities.

6 Conclusion and Future Work

Discovering communities from networks has been widely studied in recent years, which can help us to understand the latent knowledge and distributions within them. In this paper, we propose a novel community discovery model, STC, to explore communities with different topic-sentiment distributions. This model is built by combining content, links and sentiment words seamlessly, which can identify communities in a level of sentiment analysis. While most of existing methods for community identification fail to consider the valuable sentiment factor in the networks. Experimental results validated on two types of real-world datasets show that our model can detect sentiment-level communities and can achieve comparable performance, which might be applicable for the opinion analysis and decision making in large business and marketing service.

There are several future extensions to investigate for this work. The topic and sentiment words in our experiment are mixed together, it is interesting to separate them. In addition, discovering communities which have obvious sentiment differences on a certain topic is also very useful. Another direction is to investigate the evolution of communities with the change of users' sentiment topics.

References

1. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
2. Girvan, M., Newman, M.: Community structure in social and biological networks. PNAS **99**(12), 7821–7826 (2002)
3. Kim, M., Han, J.: A particle-and-density based evolutionary clustering method for dynamic networks. VLDB Endowment **2**(1), 622–633 (2009)
4. Lancichinetti, A., Radicchi, F., Ramasco, J., Fortunato, S.: Finding statistically significant communities in networks. PloS One **6**(4), e18961 (2011)
5. Li, F., Huang, M., Zhu, X.: Sentiment analysis with global topics and local dependency. In: AAAI, pp. 1371–1376 (2010)
6. Lin, C., He, Y.: Joint sentiment/topic model for sentiment analysis. In: CIKM, pp. 375–384 (2009)
7. McCallum, A., Wang, X., Corrada-Emmanuel, A.: Topic and role discovery in social networks with experiments on enron and academic email. J. Artif. Intell. Res. **30**(1), 249–272 (2007)
8. Natarajan, N., Sen, P., Chaoji, V.: Community detection in content-sharing social networks. In: ASONAM, pp. 82–89 (2013)
9. Newman, M., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004)
10. Palla, G., Barabasi, A., Vicsek, T.: Quantifying social group evolution. Nature **446**(7136), 664–667 (2007)
11. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435**, 814–818 (2005)
12. Pathak, N., DeLong, C., Banerjee, A., Erickson, K.: Social topic models for community extraction. In: The 2nd SNA-KDD Workshop, vol. 8 (2008)
13. Ruan, Y., Fuhry, D., Parthasarathy, S.: Efficient community detection in large networks using content and links. In: WWW, pp. 1089–1098 (2013)
14. Sachan, M., Contractor, D., Faruquie, T., Subramaniam, L.: Using content and interactions for discovering communities in social networks. In: WWW, pp. 331–340 (2012)
15. Tyler, J., Wilkinson, D., Huberman, B.: Email as spectroscopy: automated discovery of community structure within organizations. In: Communities and Technologies, pp. 81–96 (2003)
16. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: HLT-EMNLP, pp. 347–354 (2005)
17. Xie, J., Szymanski, B., Liu, X.: Slpa: uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In: ICDM Workshops, pp. 344–349 (2011)
18. Yang, T., Jin, R., Chi, Y., Zhu, S.: Combining link and content for community detection: a discriminative approach. In: KDD, pp. 927–936 (2009)

19. Zhou, D., Manavoglu, E., Li, J., Giles, C., Zha, H.: Probabilistic models for discovering e-communities. In: WWW, pp. 173–182 (2006)
20. Zhou, W., Jin, H., Liu, Y.: Community discovery and profiling with social messages. In: KDD, pp. 388–396 (2012)

Considerations About Multistep Community Detection

Antonio A. Gentile(✉), Angelo Corallo, Cristian Bisconti,
and Laura Fortunato

Department of Innovation Engineering, University of Salento, 73100 Lecce, Italy
antonio.gentile@unisalento.it
http://emi.unisalento.it/sna

Abstract. The problem and implications of community detection in networks have raised a huge attention, for its important applications in both natural and social sciences. A number of algorithms has been developed to solve this problem, addressing either speed optimization or the quality of the partitions calculated. In this paper we propose a multi-step procedure bridging the fastest, but less accurate algorithms (coarse clustering), with the slowest, most effective ones (refinement). By adopting heuristic ranking of the nodes, and classifying a fraction of them as 'critical', a refinement step can be restricted to this subset of the network, thus saving computational time. Preliminary numerical results are discussed, showing improvement of the final partition.

Keywords: Clustering · Community detection · Graph partitioning

1 Introduction

Network analysis has been adopted as a powerful tool in several fields related to complex phenomena [5]. Among the various strategies, outlined to understand large-scale structures, a successful one has pointed out the natural tendency of real-world networks to form *clusters*: groups of nodes densely connected among them. Even though the concept is intuitively clear, an operational definition of a 'network cluster' is itself under debate: for a concise review of suggested definitions, see [11]. Identifying these dense structures inside a network may be crucial for a wide variety of reasons.

The importance of these applications has led recently to the intense development of algorithms, aiming to solve automatically the detection of communities, or to check for the clusterability of the network [10]. The focus is here on the specific case of *community detection*, where number and size of the clusters are free parameters of the problem [24], which addresses also the issue of determining if a *good* partitioning is achievable.

On a different basis, one could distinguish among classes of algorithms, grouped according to their focus. A first class, devoted to capturing the global picture of the network clustering, aiming at a fast solution of the clustering

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 549–561, 2014.
DOI: 10.1007/978-3-319-13186-3_49

problem given, which especially suits large networks. Such algorithms will be generically indicated in the following as *coarse grain*, since in general they use global metrics as the figure of merit to optimize[1], and often embed approximated methods [4,8], thus potentially leading to a relatively high rate of misclassified nodes (e.g. see [14]). On the opposite side, fine grain algorithms, in particular those involving metrics at the node/edge level[2], or *hierarchical* structures: in this case, the aim is a precise assignment of the single nodes to the various communities. Moreover, these *refinement* algorithms frequently adopt 'exact' methods, for the optimization task they deploy.

The purpose of this paper is to provide a strategy, enabling to bridge these two different classes. At the moment, in fact, the norm is the straightforward application of a single step algorithm [10], or multi-step approaches with different optimization schedules for the same metric [23]. There is a reason behind this tendency. Small networks can efficiently rely on time consuming algorithms, thus making superfluous to adopt faster methods. These last ones are instead the only feasible chance for large networks. In this paper, we envisage that it is possible to overcome this difficulty, by running a refinement step on only a fraction of the whole network. This fraction is identified via *heuristic metrics*: we call them 'heuristic' because, as better shown in the following, the metric chosen not only draws on the characteristics of the network analyzed, but must rely on some 'preliminary' clustering results, as computed via coarse algorithms.

In Sect. 2, after a brief introduction on the framework of our proposal, we will provide a detailed assessment of general features and applicability of our multistep scheme, and discuss a few metrics which may be adopted as heuristics. Characteristics and a first testing of the method, based on heuristics proposed, will be illustrated in Sect. 3. Some remarks and outlines of future developments conclude this work.

2 Framework and Methods

In the following, we are going to use concepts and metrics derived from graph theory, assuming that:

Proposition 1. *The network to analyze can be represented by a graph $\mathcal{G}$.*

For $\mathcal{G}$ we adopt the following synthetic definition:

Definition 1. *A (directed) (weighted) graph is the ordered pair $\mathcal{G}(V, E)$, with V and E respectively the n vertices (v_i) and m edges belonging to G. If (directed), the edges $\{v_i, v_j\}$ are ordered pairs. The (weighted) values of the edges among vertices can be embedded in an 'adjacency matrix' A_{ij} of dimension n, where: $a_{ij} = 0$ iff there is no edge linking v_i to v_j.*

[1] E.g. the *optimization methods* using: *E/I ratios*, information-compression measures, ..., Hamiltonian-like quantities (spin-hamiltonians, *modularity*, ...). Another good example is the class of methods known as *block modeling* [10].

[2] Like the *edge betweenness*, *information centrality*, other cost functions, directly referred to the network structure (Kernighan-Lin approach, ...), or real-world analogies: *current-flow*, *message-passing*, ...

$\mathcal{G}$ may be a *digraph*[3]. This case is explicitly analyzed in the following, where the ordering in the indexes of matrix A (in this case non-symmetric) will be supposed to follow the rule: edge $i \to j$ is embedded in the element a_{ij}, and viceversa. Also the case of a *multimodal graph* can be treated in principle, supposing that a clustering problem, as in Definition 2, is well posed for the graph considered.

As pointed out in the introduction, we intend to address the general problem of automatic clustering in a network. Therefore, we will mainly refer to:

Definition 2. *'Community detection' as the optimal partition problem*[4] *of finding* $\mathcal{C}_k$ *non-overlapping, non-empty components (i.e. subgraphs) of* $\mathcal{G}$*:* $\bigcup_k \mathcal{C}_k = \mathcal{G}$*. Their number* k *may be an input of the problem, or left as a free parameter.*

Proposition 1 and Definition 2 are strictly required, for the following discussion to make sense. The assumption of 'no-overlap', instead, may be (partially) relaxed, though leading to interesting applications. Moreover, if the partitions $\mathcal{C}_k$ optimize a 'quality function'[5], it would be eased a quantitative comparison among different solutions (eventually found at different steps, or by different combinations, of the global scheme as in Fig. 1). Further comments can be found in [12].

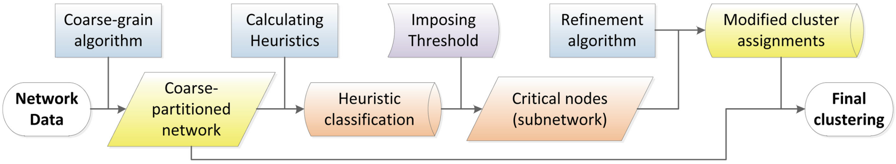


Fig. 1. Flowchart of the multi-step method proposed in the text. In blue/violet are distinguished the operational steps. Other colors are referred to generic data (color figure online).

2.1 The Multi-step Scheme

Our contribution for a multi-step approach is the proposal of *heuristic metrics*, that have both low computational time-complexity, and a good efficiency in classifying the nodes according to their degree of membership to possible communities. These metrics enable the adoption of a scheme including:

1. a *coarse grain algorithm* for the initial clustering guess,
2. an efficient, *heuristic metric* for the retrieval of a reduced set of nodes, requiring further analysis upon cluster assignment,
3. a *refinement algorithm*, to be run on the nodes produced by the previous step, so to improve the 'quality' of the final partitions.

[3] For a detailed review of peculiarities of clustering approaches in directed graphs, the interested reader may refer to [21].

[4] We stress not to confuse it with the *graph partitioning*, i.e. a specific clustering problem.

[5] Which may equivalently be a 'cost function', for a survey see [10].

These three main elements, along with some other features which will be introduced in the text, are in Fig. 1, which shows the global multi-step structure.

The very same introduction of a *refinement* brings along the problem of identifying a measure, able to compare different clustering solutions (i.e. a *relative* measure). A general discussion about the problem is clearly outside our scope: additional information can be found in [15,18,26].

In the following, the ultimate target will be to approximate those partitions, which would be provided if the fine-grain analysis chosen was to be performed on the whole network (implicitly assumed to be the best partitioning available). About the distance among partitions provided at different steps in our procedure, we adopt the ratio between the minimum number of elements to delete from a graph $D(P, P')$, so that the two induced partitions become identical, and the size of the graph [12]: $D(P, P') := n_D(P, P')/n$.

An effective multi-step procedure requires a few qualitative hypotheses:

Proposition 2. *refinement algorithm used must be able to perform displacements of single nodes;*

Proposition 3. *the heuristic metric chosen must perform as a good figure of merit, in quantifying the 'criticality' of the nodes in the network;*

Proposition 4. *however chosen, the fastest method (eventually approximate) to compute the metric in Proposition 3 should at least outperform, in time-complexity, the refinement clustering algorithm.*

Proposition 2 derives from the necessity, once a node-ranking has been established, to analyze, and eventually modify, the cluster attribution of specific nodes, so to address the way the refinement algorithm works. The second statement emphasizes how a perfectly efficient metric should rank first *only* those nodes which will be misassigned by the coarse grain algorithm. Clearly, given that a variety of algorithms could be used as coarse-grain, this 'perfect efficiency' is indeed a relative concept, and independently from the refinement algorithm there is no way to define it. Finally, for an alternative clustering procedure to be competitive, with respect to the refinement algorithm, all of its steps must be (much) faster to compute, as stated in Proposition 4.

A first naive approach, for retrieving the critical nodes of the network, could be to adopt *centrality* measures from network theory. However, there are a few drawbacks [12], such as the implicit assumption, that the refinement should involve the most 'important' nodes. Misclassifying a central node is likely more problematic, but there is no general reason why the coarse-grain algorithm should perform worse on most central nodes.

Let us introduce a few qualitative statements, aiming to satisfy the requirements in Propositions 2–4. As the first, the assignment of a node to a cluster depends the distribution of its links to neighbour nodes [10]: this leads to introducing the node degrees. In order not to relate the heuristic to the importance of the node, some normalization factor must be introduced. In undirected graphs, we will use a total 'symmetrized' degree for each node j, $d_T(j) := \sum_i (a_{ij} + a_{ji})/2$.

Given the hypothesis of computing heuristics only after a first coarse assignment of nodes to clusters, one is able to distinguish among edges *inside* or *outside* a given cluster, via the binary function *com* with values in $\{-1, 1\}$:

$$com(i,j) := \begin{cases} -1 & \text{(if } i \text{ and } j \text{ belong to different communities)} \\ +1 & \text{(if } i = j \vee \text{ if } i \text{ and } j \text{ belong to same cluster)} \end{cases} \tag{1}$$

We claim that a 1^{st} order heuristic metric, suitable for quantifying the criticality of node j, can be formulated as:

$$H_1(j) = \frac{1}{2d_T(j)} \sum_i (a_{ij} + a_{ji})\, com(i,j) \tag{2}$$

while for the 2^{nd} order heuristic we suggest:

$$H_2(j) = \frac{1}{2d_T^2(j)} \sum_{i \neq j} (a_{ij} + a_{ji})\ \ com(i,j) Q\ d_T(i) H_1(i) \tag{3}$$

$$Q := \frac{\delta(\mathcal{G})}{\Delta(\mathcal{G})} \tag{4}$$

is a normalization factor, with $\delta(\mathcal{G})$ and $\Delta(\mathcal{G})$ the minimum and maximum degree of the nodes in $\mathcal{G}$, respectively.

A few remarks. The expressions about the *order* refer to the width of the network sample taken into account for each node: edges shared *with* its neighbour nodes in the 1^{st} case, and also all edges shared *by* its neighbour nodes in the 2^{nd}.

Both heuristics are bounded: as it is easy to verify, $-1 \leq H_1,\ H_2 \leq +1$. Thus, the first order heuristic may be interpreted as a normalized measure of the *correlation* of the node with its cluster of assignment, disregarding its neighbour nodes. Evidently, a positive correlation is here an index of robust assignment, whereas negative correlations indicate misassignment.

Qualitatively, re-introducing in (3) the heuristic H_1 accounts for the cluster assignment of neighbour nodes: the stronger the connection of a neighbour node i to its own cluster, the higher we expect its contribution to the (mis)assignment score of analyzed node j, if $com(i,j) = +1$ (-1). The factor

$$M := Qd_T(i)/d_T(j), \tag{5}$$

instead, can be interpreted as a measure relating the contribution from node i to its relative 'importance' in the network, compared to node j (thus the presence of Q). That is, M reduces the contribution from H_2, compared to H_1: if $d_T(i)/d_T(j) = \rho \Rightarrow M < \rho^2$.

Another interesting point to analyze is how to combine the two heuristics. We suggest, as the most profitable figure of merit, the convex combination:

$$H(j) := \alpha H_1(j) + (2 - \alpha) H_2(j) \tag{6}$$

with $\alpha \in [0, 2]$. In the following, illustrating the proposal, we will restrict considerations to the simplest case with $\alpha = 1$.

It is worth to comment how the introduction of heuristics as above may be regarded as a 'mean field like' procedure, where only pairwise, nearest neighbour interactions are considered (which is the case, for example, in Ising models). The quantity H itself can be interpreted as a *potential*, once changed in sign. One may notice that procedures based on optimization of *Hamiltonians* have already been thoroughly applied to the clustering problem (e.g. [20,25]). Indeed, with a terminology drawing on this parallel, a key difference in our approach is that we are defining and using *local* potentials, whereas the traditional approach involves the optimization of a *global* potential.

2.2 Further Comments on the Heuristics

Given that the heuristics, in the form introduced so far, were only intuitively justified to be reliable metrics for our aim, it is plenty of possible modifications, simplifying or generalizing the particular version given in (2) and (3).

We will take in consideration a few cases which may be interesting for some particular applications. As the first, whenever a speed-up in the computation of the heuristics is required, it is envisaged the possibility to slightly change the definition of *com* (1), so to skip operations on *positive* (or, equivalently, *negative*) terms. Therefore, this version of the algorithm could use e.g.:

$$com^{+}(i,j) = \begin{cases} 0 & \text{(if } i \text{ and } j \text{ belong to different communities)} \\ +1 & \text{(if } i = j \vee \text{ if } i \text{ and } j \text{ belong to same cluster)} \end{cases} \tag{7}$$

or viceversa for $com^{-}(i,j)$. Steps involving null terms in the computation of H_1 and H_2 would be excluded by conditional restraints.

A more interesting case is given by *directed* graphs (i.e. 'digraphs'). In fact, to keep the general case as simple as possible, we have always avoided directionality considerations in (2) and (3), by using the averaged term $(a_{ij}+a_{ji})/2$. Intuitively, this is equivalent to the replacement of multiple directed (weighted) edges, for each couple of nodes, with a single undirected weighted edge. Even if approaches like this have been applied to highly successful analyses of naturally directed graphs [1], it is well recognized how intrinsic directional features may add insight to static [6] or dynamic [17] analyses of networks. Notice that the heuristics introduced may be readily generalized to include different expressions for an 'in-metric' $H^{in}_{1,2}$ as well as an 'out-metric' $H^{out}_{1,2}$. I.e. for the inner case:

$$H^{in}_1(j) = \frac{1}{d^{in}_T(j)} \sum_i a_{ij}\ com(i,j) \tag{8}$$

$$H^{in}_2(j) = \frac{1}{[d^{in}_T(j)]^2} \sum_{i \neq j} a_{ij}\ com(i,j) Q\ d^{in}_T(i) H^{in}_1(i) \tag{9}$$

Now, considerations about robustness of products of inner and outer quantities, for clustering procedures, may apply to this case. In fact, multiplying H^{in}_1 and

H_1^{out}, the product (H_1') closely resembles[6] the *vertex-cluster affinity*, employed in [27] for the *graph degree-linkage* method, where the cluster would here be the neighborhood $\mathcal{N}$ of each critical vertex. The contribution from H_2' can instead be seen as an improvement of this affinity. Therefore, drawing on these previous results, we claim that a robust implementation of our procedure in directional cases uses the node heuristics:

$$H_1'(j) := H_1^{in} H_1^{out} \tag{10}$$

$$H_2'(j) := H_2^{in} H_2^{out} \tag{11}$$

and the obvious generalization of (6) for their combination.

It is still left open, the possibility to drastically change the form of the heuristic. For example, given that H_1 is claimed to be a measure of the membership degree of node j to its initial community, one could recall how this indication is embedded in the elements of the *membership matrix*, as defined in [3]. However, the additional definitions of 'positions' and 'distances' in a metric space, required in the definition of this matrix, may be rather artificial for some graphs [10].

Again modifying preliminary definitions: Q, as given in (4), may be considered a rough figure of merit for the degree ratio in (5). E.g. one could assume a Gaussian behaviour in the degree distribution, and thus suppose Q to be in the form of a standard deviation[7]:

$$Q^2 = \frac{2}{n(n-1)} \sum_{\substack{i,j \\ i \neq j}} [\delta_D(i,j)]^2, \tag{12}$$

with $\delta_D(i,j) = (d_T(i) - d_T(j))$. Therefore, it may be objected that:

$$M' := exp(-\delta_D(i,j)^2/Q^2) \tag{13}$$

is a more reliable measure as a *degree distance* among nodes i and j and should replace M (5) in (3). Notice that (13) provides a measure, on the strength of the connection between the nodes, resembling the *dimensionality reduction* procedure invoked for graph construction in [2] and related works. However, two considerations hold. The first and more important is that computing the quantity in (12) is a computational problem much more costly (and in some cases even tricky [7]), than the linear scan required for computing (4). A second noticeable problem is that the naive introduction of this 'standard variance' form for Q does not fit well our requirements. Indeed, it is introduced an unwanted symmetry: M' is a factor reducing the importance of H_2, indifferently of whose node is the degree centrality increasing. In formulas, where ϵ is $O(e^{-n^2})$:

$$lim_{d_T(j)/d_T(i) \to \infty} M \longrightarrow 0 \tag{14}$$

$$lim_{|d_T(j) - d_T(i)| \to \infty} M' \longrightarrow \epsilon \tag{15}$$

[6] The similarity of these quantities does not imply similarity in their usage, as in [27] the vertex-cluster affinity (and its derivatives) are directly used for the agglomerative step of the algorithm, whereas we use them only to classify the quality of single-node attributions to clusters.

[7] Notice that the expected value for the population is trivially $< \delta_D >= 0$.

There are certainly various possibilities to solve the issue: e.g. introducing further parameters in δ_D, or defining it differently. However, in our opinion this unnecessarily complicates the global picture, and therefore move on to test numerically the performance of the heuristics outlined.

Table 1. Analysis of computational complexity (average, per node) and memory usage (globally) of the quantities involved in the calculation of the heuristics in Eqs. (2) and (3), for an undirected graph. *c.g.?* indicates that the complexity of this step depends on the coarse algorithm applied. $com(i,j)$ is supposed to be retrieved from a stored vector of single-node assignments.

	$H_1(i)$	$d_T(i)$	$com(i,j)$	Q	$\dots \sum_i (a_{ij} + a_{ji}) \dots$	*Total*
$H_1(i)$	-	$\mathcal{O}(m/n)$	c.g.?	-	$\mathcal{O}(m/n)$	$\mathcal{O}(m/n)$
$H_2(i)$	$\mathcal{O}(m/n)$	$\mathcal{O}(m/n)$	c.g.?	$\mathcal{O}(1)$	$\mathcal{O}(m/n)$	$\mathcal{O}(\frac{m}{n} + 1)$
size	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(m)$	$\mathcal{O}(m + n)$

2.3 Discussion About Implementation

We are now left with checking the respondency of our proposal for a heuristic, to the requirements stated in Pts. 2–3 of Proposition 2. Observing Table 1, it is easy to see that H_1 has a complexity of $\mathcal{O}(m)$ and H_2 has a complexity of $\mathcal{O}(m + n)$, under the following assumptions: the graph is undirected and stored as an ordered *edgelist*[8], coarse communities have already been calculated and stored in a vector. Notice how redundant terms in the two heuristics can ease the subsequent calculation of both quantities $H_{1,2}$. Such a complexity is a reasonably good result: one of the fastest coarse algorithms for community detection runs with complexity $\mathcal{O}(n + m)$ on sparse graphs. Additionally, operations leading to the heuristics' complexity are very basic, thus we envisage very low factors.

In order to perform a test for the multi-step scheme, following also Fig. 1, two elements are required to be explicitated.

A *coarse grain algorithm* for the first step. We chose to use the *fast Newman* (FN) approach [22] with a greedy modularity optimization of the *modularity*, as suggested in [8]. Within this implementation, it is known to run in $\mathcal{O}(n\ log^2 n)$ on sparse graphs. This method is of widespread adoption in the literature[9] and in several network analysis softwares.

A *refinement algorithm* for the final step. In this case we introduced a modified *Girvan-Newman* (GN) method, based on the *edge betweenness*: a perfect example of an algorithm unfeasible to be used straightforward for large networks, as it requires $\mathcal{O}(n^3)$ time (sparse case). The original version of this algorithm was not intended to perform single node re-assignments [13], so that it is here modified, even though keeping the same local measure as the working principle. In brief, here the edge betweenness is calculated only for *critical edges*, i.e.

[8] If not, an additional step with complexity $\mathcal{O}(m\ log\ m)$ must be taken into account.

[9] Its combined simplicity and robustness make the FN method very popular, even if several works have started to point out its ineffectiveness for specific cases [14,16].

those edges linking couples of nodes, of whose at least one is *critical*. The last edge to be removed, before a node is isolated, is also the one ruling the community assignment[10]. Notice that the refinement algorithm used is allowed both to eventually shrink the number of clusters composing the final partitioning, *and* to create new clusters, eventually not resolved by the coarse step.

The adoption of a refinement step poses a non-trivial problem: given unawareness of the percentage of nodes classified in the wrong cluster by the coarse algorithm, how many nodes must be 'refined' analyzed, among those scoring worse in H? That is, we need to impose a threshold to the heuristics (see Fig. 1), selecting as critical nodes only those having a lower value of H. In our opinion, this point requires a good insight about the structure of the network, and if investigated, can provide interesting results. A pragmatic and prudential solution is: pose the threshold in H as high, as the additional computational time, required for refinement, is considered feasible by the adopter. For numerical tests below, we will adopt instead an 'absolute' approach: the refinement algorithm will be run on all, and only, those nodes having negative values of H.

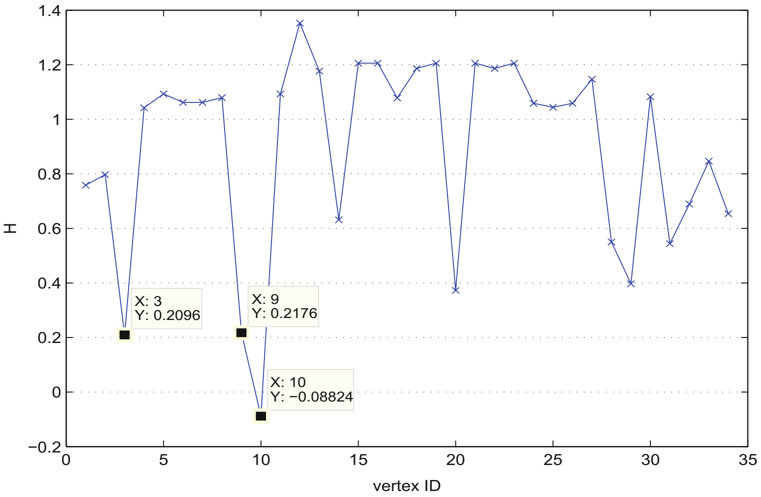


Fig. 2. H (Eq. 6) for the Zachary's karate club case. Datatips are displayed in the first case for those vertices exhibiting the lower scoring, and therefore the most critical ones. Vertices IDs are the same as in [13].

[10] Specifically, we progressively remove critical edges with high edge-betweenness. The algorithm has three hierarchical rules to assign node to the refined *community vector*: (i) if a critical node i is left with one only edge linking it to a non-critical node j (i.e. $a_{ij} + a_{ji} \neq 0$), i acquires the same community assignment of j: $i \in \mathcal{C}_k$ *iff* $j \in \mathcal{C}_k$; (ii) for critical nodes pointing to each other, before becoming isolated by the edge-removal procedure ('queued nodes'), it is attempted the creation of a new community; (iii) if this attempt fails, the *transitivity principle* introduced in the text is used to infer the non-critical node ruling the community assignment.

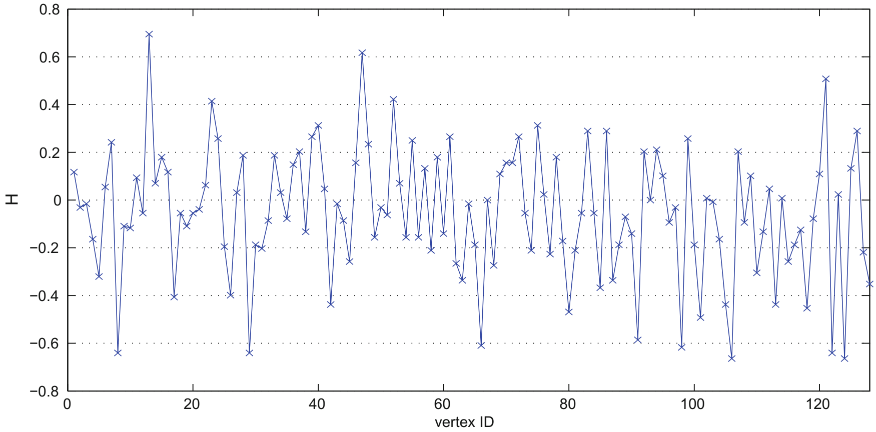


Fig. 3. Combined heuristic H for the *Girvan-Newman benchmark* cited in the text. Generated using the code as in [19], with average degree $\bar{d} = 16$, and mixing parameter $\mu = 0.6$.

3 Preliminary Tests

This paragraph is devoted to show how the particular implementation of a multistep scheme (as outlined in Sect. 2.3) works for a real case, and in particular to test if the heuristics, introduced so far, are capable of satisfying the requirements stated at the beginning of this section.

Test-cases. We have chosen to focus on the split of a *karate club* in two different 'communities', studied in [22]. This example fits well a preliminary, qualitative discussion, because it is small enough ($n = 34$) to let us follow in detail the performance of the heuristics[11]. In Fig. 2 is the sum of the heuristics H_1 and H_2 for all the vertices of the karate club network, after a coarse assignment of clusters has been performed through the application of the FN algorithm. It is evident how almost all of the nodes have positive values of both $H_{1,2}$. This confirms that H captures the good performance of the FN algorithm in this test. We can also state that our core claim is satisfied: the node #10, known to be misclassified by the coarse algorithm [22], is the one scoring worse, and even has a negative H, as shown in Fig. 2. Notice also that the GN refinement correctly classifies this node, displacing it into the 'right' cluster. Recalling $D(P, P')$, the coarse method has in this case a distance of $D \cong 0.029$ from the partition found by our scheme: this distance can be understood as the *improvement* provided for the solution[12].

[11] Yet it is complex enough to pose difficulties for the fast coarse algorithm chosen [24].

[12] Noticeably, for this specific case, the GN method is known to classify incorrectly node #3 [13], which actually ranked worst, immediately after node #10, in the H scoring. This further suggests how the heuristic proposed is indeed efficient, in sorting nodes with uncertain cluster assignment.

As a counter-example, in Fig. 3 we report the heuristics, calculated after the same coarse step, run on an artificial network[13] with $n = 128$. This network resembles a 'Girvan-Newman benchmark' with communities of variable size, and with parameters which are known to make the community assignment fail, when performed by the FN algorithm [9,19]. It is immediately evident how the average scoring of the heuristic H is much worse than the previous case, and how most of the nodes exhibit negative scoring. This indicates again that the heuristics scouts nodes misclassified in the coarse step.

4 Conclusions

Summarizing the main results of this work: we have proposed the adoption of a multi-step scheme, to improve the results of clustering algorithms, with a particular focus on community detection. This scheme basically includes: the adoption of a (state-of-art) fast, coarse algorithm for the first step; an accurate refinement algorithm, specifically adapted for this purpose; to bridge these two elements, a novel set of heuristic metrics. These last ones are the core of the proposal: they are intended to scout those nodes potentially tricky in the cluster assignment, and thus worth to be analyzed by the refinement step. We have shown, with the aid of test-cases, that the heuristic introduced satisfies the requirements of being computable with low time-complexity, and may efficiently retrieve those nodes which turn out to 'deceive' the less accurate algorithms.

In future developments there is the plan to systematically investigate to what extent our approach reveals useful for application to real world and computer generated networks (thus identifying its limits). In particular, the aim will be about large scale networks, for which it may also be unknown the 'true partitioning' (whether obtained via a direct observation, or as provided by the application of the refinement to the whole network). In this case the only possible check would be the comparison with results, as provided by different fast algorithms. Another direction, for further analyses, is given by the limitations already found for modularity-based approaches [14]: we claim that our multi-step strategy may (partially) solve the degeneracies displayed by these approaches for particular cases. Verification of this conjecture could lead to important applications.

References

1. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**(5439), 509–512 (1999)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. **15**(6), 1373–1396 (2003)
3. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers, Norwell (1981)

[13] Created through the benchmark package available at: goo.gl/Btp70b.

4. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech: Theory Exp. **2008**(10), P10008 (2008)
5. Brandes, U., Erlebach, T. (eds.): Network Analysis: Methodological Foundations. LNCS, vol. 3418. Springer, Heidelberg (2005)
6. Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., Wiener, J.: Graph structure in the web. Comput. Netw. **33**(1), 309–320 (2000)
7. Chan, T.F., Golub, G.H., LeVeque, R.J.: Algorithms for computing the sample variance: Analysis and recommendations. Am. Stat. **37**(3), 242–247 (1983)
8. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**(6), 066111 (2004)
9. Danon, L., Díaz-Guilera, A., Arenas, A.: The effect of size heterogeneity on community identification in complex networks. J. Stat. Mech: Theory Exp. **2006**(11), P11010 (2006)
10. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3), 75–174 (2010)
11. Fortunato, S., Latora, V., Marchiori, M.: Method to find community structures based on information centrality. Phys. Rev. E **70**(5), 056104 (2004)
12. Gentile, A.A., Corallo, A., Bisconti, C., Fortunato, L.: Proposal for heuristics-based refinement in clustering problems. In: MMB & DFT 2014, Proceedings of the International Workshops, pp. 19–34. University of Bamberg Press (2014)
13. Girvan, M., Newman, M.E.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. **99**(12), 7821–7826 (2002)
14. Good, B.H., de Montjoye, Y.A., Clauset, A.: Performance of modularity maximization in practical contexts. Phys. Rev. E **81**(4), 046106 (2010)
15. Kannan, R., Vempala, S., Vetta, A.: On clusterings: good, bad and spectral. J. ACM (JACM) **51**(3), 497–515 (2004)
16. Kehagias, A.: Bad communities with high modularity (2012). arXiv:1209.2678
17. Krapivsky, P., Rodgers, G., Redner, S.: Degree distributions of growing networks. Phys. Rev. Lett. **86**(23), 5401 (2001)
18. Kriegel, H.P., Pfeifle, M.: Measuring the quality of approximated clusterings. BTW **5**, 415–424 (2005)
19. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E **78**(4), 046110 (2008)
20. Liu, S., Ying, L., Shakkottai, S.: Influence maximization in social networks: an ising-model-based approach. In: Communication, Control, and Computing, pp. 570–576. IEEE (2010)
21. Malliaros, F.D., Vazirgiannis, M.: Clustering and community detection in directed networks: a survey. Phys. Rep. **533**(4), 95–142 (2013)
22. Newman, M.E.: Fast algorithm for detecting community structure in networks. Phys. Rev. E **69**(6), 066133 (2004)
23. Newman, M.E.: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E **74**(3), 036104 (2006)
24. Newman, M.E.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. **103**(23), 8577–8582 (2006)
25. Reichardt, J., Bornholdt, S.: Detecting fuzzy community structures in complex networks with a potts model. Phys. Rev. Lett. **93**(21), 218701 (2004)

26. Robardet, C., Feschet, F., Nicoloyannis, N.: An experimental study of partition quality indices in clustering. In: Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, PKDD '00, pp. 599–604. Springer, London (2000)
27. Zhang, W., Wang, X., Zhao, D., Tang, X.: Graph degree linkage: agglomerative clustering on a directed graph. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 428–441. Springer, Heidelberg (2012)

An Edge-Centric Approach for Change Point Detection in Dynamic Networks

Yongsheng Cheng(✉) and Xiaokang Lin

State Key Laboratory on Microwave and Digital Communications,
Tsinghua National Laboratory for Information Science and Technology,
Department of Electronic Engineering, Tsinghua University,
Beijing 100084, China
chyongsheng@gmail.com

Abstract. The graph-theoretic analysis of dynamic networks has attracted much research interests recently. Change point detection is essential to understand the dynamic structure of time evolving networks. This work proposes an edge-centric approach to detect the change points of dynamic networks. In the proposed method, a singular value decomposition (SVD) is performed on a newly defined edge-segment matrix and the decomposition is projected to a lower dimensional latent space. Then the dissimilarity between graph segments is calculated for detecting the change points. The approach applies to directed/undirected and weighted/unweighted dynamic graphs. Experiments are conducted on both a synthetic dataset and the Enron email dataset. Results show that change points of the dynamic networks are effectively detected by the proposed approach.

Keywords: Change point detection · Dynamic networks · Graph segments · Latent semantic analysis

1 Introduction

Research on graphs naturally arises in the study of social networks, sensor networks, and citation networks, etc. In a graph formulation, nodes represent individual objects, and edges represent relationships and interactions among these objects. The majority of existing studies assume the underlying graph is static. That is, the topology of the network remains unchanged over time. However, most of the real networks are dynamic and exhibit structural changes over time. The changes are expressed by dynamic relationships among nodes in the network. Dynamic networks are attracting increasing interest due to their potential in capturing natural and social phenomena over time. An example of dynamic networks is the proximity based mobile social network [1], where the connections among the objects depend highly on the physical proximity and change constantly over time.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 562–571, 2014.
DOI: 10.1007/978-3-319-13186-3_50

An inherent problem in dynamic networks is change point detection. This is a new challenge in graph mining, where one needs to discover the unknown structures hidden deep inside of the graphs, and detect their anomalous changes. A change point always corresponds to an anomaly in networks. Therefore, change point detection helps to understand the anomalous behaviours or faults in dynamic networks [2]. In the public safety case, change point detection of a mobile social network may assist to identify terrorist activities [3]. Moveover, the detection of change points is an important step to summarize the network activity with a fewer number of static graphs [4], which can be used to reduce the amount of data for representing a highly dynamic network.

This work proposes an edge-centric change point detection algorithm. In this algorithm, a new edge-segment matrix is used. The algorithm is analogous to latent semantic analysis (LSA) [5], which is a technique in natural language processing for analyzing relationships among a set of documents and the terms. Using the proposed method, the change points of the latent structures are detected by evaluating the dissimilarities between consecutive graph segments. In this work, the terms "network" and "graph" are used interchangeably.

The rest of this paper is organized as follows. Related works are briefly reviewed in Sect. 2. The edge-centric algorithm is proposed in Sect. 3. In Sect. 4, experiments with synthetic and real datasets are provided, which demonstrate the effectiveness of the proposed algorithm. Finally, the conclusion and future work are provided in Sect. 5.

Notations. In the sequel, vectors are denoted by boldface lower-case letters, and matrices by boldface upper-case letters. For a matrix $\mathbf{A}$, $\mathbf{A}^{'}$ is the transpose. Let $\text{vec}(\mathbf{A})$ denote the vector of columns of $\mathbf{A}$ stacked one under the other.

2 Related Work

The structure of a dynamic network may change considerably between consecutive snapshots. Change point detection is a form of anomaly detection, which always arises in dynamic analysis.

Some previous studies in dynamic networks adopt a two-step approach [6,7] for mining the network structure. First, static community detection algorithms, e.g., modularity maximization [8], is applied to each snapshot of the network at different time steps. Second, community evolution is introduced to interpret the change of communities over time. The work in [9] considers the smoothness of the structure between consecutive snapshots. The authors propose a formulation which automatically provides a trade-off between the accuracy of the clustering obtained, and the deviation from one time step to the successive. In highly dynamic networks where not all of the nodes belong to reasonable communities, the static community detection for each snapshot may not result in meaningful outcome. Therefore, the two-step and smooth approach is not applicable to such cases.

From another perspective, many heuristic statistics have been proposed to capture the community variation. Change detection is then performed by comparing the statistics to a predefined threshold. In [2], the angle of the principal eigenvector is tracked in a dynamic computer system. An anomaly is declared if the angle changes by more than some threshold. Authors in [4] define an average distance between consecutive graph snapshots based on nodes' connectivity. Then they propose to detect the change points with this distance measure. In [10], scan statistics, which capture the history of a node's neighborhood, are introduced to detect anomalous behaviors. In [11], the authors propose to use a fusion of nine different statistics to detect anomalies and change points. The work in [12] introduces a parameter free method based on information theoretic principles to find community structure and change points. Authors of [13] consider an anomaly score based on the eigen decomposition of the adjacency matrix. Then a statistical test based on randomized power martingale is used to detect the change points. Although the above detection statistics are intuitively interpretable, some of them are rather complex. Besides, algorithms in [2,4,11,13] are designed for undirected networks, and algorithms in [10–12] are designed for unweighted networks. The work in [14] considers dynamic weighted directed graphs. The detection algorithm compares a simple similarity measure to a predefined threshold. However, as shown in [14], the performance is severely influenced by the choice of the threshold.

This work proposes a novel edge-centric approach for change point detection in dynamic networks. The approach is based on the singular value decomposition (SVD) of the edge-segment matrix. It is able to detect the change points of the latent structures in directed/undirected and weighted/unweighted dynamic networks.

3 Proposed Method

In this section, the edge-segment matrix is defined. Afterwards, an edge-centric change point detection algorithm is proposed.

3.1 Edge-Segment Matrix

A dynamic graph is a sequence of random graphs denoted by $\mathcal{G} = \{G^{(1)}, G^{(2)}, \cdots, G^{(T)}\}$, where $T < \infty$ is the length of the sequence. Each snapshot $G^{(t)} = (V^{(t)}, E^{(t)})$ is a static graph with vertex set $V^{(t)}$, edge set $E^{(t)}$ and adjacency matrix $\mathbf{A}^{(t)} = [A^{(t)}_{i,j}]$. Without loss of generality, it is assumed that the vertex set $V^{(t)}$ is the same for all graph snapshots; otherwise, one can introduce all-zero rows and columns into the adjacency matrices.

A graph segment [12] is a set of consecutive graphs $\mathcal{G}^{(s)} = \{G^{(t_s)}, G^{(t_s+1)}, \cdots, G^{(t_{s+1}-1)}\}$, where $t_s < t_{s+1}$. Define $GS^{(s)} = \bigoplus_{t=t_s}^{t_{s+1}-1} G^{(t)}$ as a cumulative graph for the s-th graph segment. The corresponding adjacency matrix is defined as $\mathbf{AS}^{(s)} = [AS^{(s)}_{i,j}]$, where $AS^{(s)}_{i,j} = \sum_{t=t_s}^{t_{s+1}-1} A^{(t)}_{i,j}$.

Define $\mathbf{e}^{(s)} = \mathrm{vec}(\mathbf{AS}^{(s)})$. That is, $\mathbf{e}^{(s)}$ is a vector containing the weights of all the edges in the cumulative graph $GS^{(s)}$. For undirected graphs, only the upper (or equivalently, the lower) triangular part of the adjacency matrix $\mathbf{AS}^{(s)}$ is considered.

Now, the edge-segment matrix can be defined as

$$\mathbf{E} = [\mathbf{e}^{(1)}, \mathbf{e}^{(2)}, \cdots, \mathbf{e}^{(T_{\mathrm{s}})}] \,, \tag{1}$$

where T_{s} is the number of graph segments. Generally, for the purpose of a fair comparison between different graph segments, it is assumed that each segment is of equal length. The length can be chosen appropriately according to the size of datasets and the requirement of applications.

3.2 Edge-Centric Change Point Detection Algorithm

Notice that the edge-segment matrix is analogous to the term-document matrix in LSA. The weight $E_{i,j}$ corresponds to the term frequency of the i-th term (edge) in the j-th document (graph segment). As in LSA [5], by performing an SVD on the edge-segment matrix, and projecting the decomposition to a lower dimensional latent space, then the dissimilarity between two graph segments can be captured by the cosine distance between the corresponding feature vectors. Explicitly, the SVD of $\mathbf{E}$ is expressed as,

$$\mathbf{E} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}' \,.$$

The elements of the diagonal matrix $\mathbf{\Sigma}$ are called singular values and the columns of $\mathbf{U}$ and $\mathbf{V}$ are called left and right singular vectors, respectively. Let $\mathbf{\Sigma}_k$ denote the reduced matrix with only the highest k singular values. Also, $\mathbf{U}$ and $\mathbf{V}'$ are reduced to $\mathbf{U}_k$ and $\mathbf{V}'_k$ to have k dominant columns and rows, respectively. Therefore, a reduced-rank approximation of $\mathbf{E}$ is obtained as,

$$\mathbf{E}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}'_k \,.$$

This corresponds to projecting the vector representation of each edge and graph segment into a k-dimensional subspace whose axes form k latent directions. The resulting reduced-dimension representation $\mathbf{E}_k$ is the best rank-k approximation to the original matrix in the least-squares sense [15].

The graph segments are now characterized by the column vectors of $\mathbf{\Sigma}_k\mathbf{V}'_k$ [5]. The relation between two graph segments can be described by the cosine distance between their corresponding k-dimensional representations,

$$d(i,j) = 1 - \frac{\mathbf{q}'_i\mathbf{q}_j}{|\mathbf{q}_i||\mathbf{q}_j|} \,, \tag{2}$$

where $\mathbf{q}_i$ is the i-th column of $\mathbf{\Sigma}_k\mathbf{V}'_k$. This is the key idea that we propose to detect the change points in dynamic graphs. Intuitively, if the network structure does not change much over time, consecutive graph segments of the dynamic graphs have similar descriptions and a small cosine distance. Whenever a graph segment changes severely with respect to previous ones, a large distance is caused.

Based on the cosine distance between consecutive graph segments, change point detection can be performed by comparing the distance to a pre-defined or dynamic threshold. For the latter, typical mean-standard deviation based methods are readily applicable. For example, define $\mathcal{D}_t = \{d(1,2), d(2,3) \cdots, d(t-1,t)\}$ to be the set of cosine distances calculated by (2) up to time t. The dynamic threshold can be defined as,

$$d_t^{\text{th}} = \text{mean}(\mathcal{D}_t) + \alpha \cdot \text{std}(\mathcal{D}_t) ,$$

where α is a positive number; mean($\cdot$) and std($\cdot$) denote the mean and standard deviation of a dataset, respectively. Parameter α can be chosen empirically according to the number of change points to be retrieved. At the t-th step, the detection algorithm compares $d(t,t+1)$ to the threshold d_t^{th}. Whenever $d(t,t+1) > d_t^{\text{th}}$, a change point is declared; moreover, in the following steps, $d(t,t+1)$ is set to be mean($\mathcal{D}_t$) for calculating the dynamic threshold.

Also, as in LSA, the edge-segment matrix can be preprocessed by local and global weighting before the SVD operation for improving the retrieval performance. Empirical studies report that the Log-Entropy weighting functions work well in practice [16,17]. In the Log-Entropy weighting scheme, the local weighting

$$l_{i,j} = \log(E_{i,j} + 1)$$

is applied to each entry in the matrix. The global weighting

$$g_i = 1 + \sum_j \frac{p_{i,j} \log p_{i,j}}{\log T_\text{s}} ,$$

where $p_{i,j} = \frac{E_{i,j}}{\sum_j E_{i,j}}$, is applied to row i of the matrix.

Then, each entry $\hat{E}_{i,j}$ of the weighted edge-segment matrix $\hat{\mathbf{E}}$ is computed as,

$$\hat{E}_{i,j} = g_i l_{i,j} .$$

3.3 Discussions

In dynamic graphs, edges are characterized by their appearance in the graph segments; graph segments are characterized by the edges that they contain. Using the above proposed method, the edges and graph segments are now represented as vectors in the k-dimensional space. Through dimension reduction, the method extracts the most significant features of the dynamic graphs. In literature, it is shown that for a fairly large range of values of k, the reduced dimensional approach performs substantially better than computing the similarity directly between two graph segments [18].

Recall that in LSA, not only does a query term match documents that contain it, but it matches documents that contain similar terms as well. In dynamic networks, an edge does not necessarily appear in every graph segment. But by performing similar operations as in LSA on the edge-segment matrix, an edge can still have a strong relation with some segments that do not contain it, as long as some strongly related edges appear in the segments. That is, this method focuses on the latent community structure instead of on individual edges. Therefore, this method potentially uncovers the dynamic latent structures embedded in the edge-segment matrix.

The proposed method is simple and elegant in that the most expensive step only requires a partial SVD operation to compute the first k singular values and singular vectors. Analogous to LSA, the order of edges is unimportant in the edge-segment matrix for change point detection. Moreover, an edge which never appears–corresponding to an all-zero row vector in the matrix, can be removed to reduce the dimension of the problem. Since the edge-segment matrix is always sparse, the partial SVD can be computed efficiently through sparse iterative algorithms, for example, the Arnoldi [19] and the Lanczos [20,21] algorithms. In the case that the detection needs to be performed dynamically and sequentially, incremental SVD algorithms [22] can be utilized.

Moreover, the proposed approach is also general enough, since it can be applied to directed/undirected and weighted/unweighted dynamic graphs. As will be shown in the following section, this edge-centric approach effectively detects the change points in synthetic and real dynamic networks.

4 Experimental Evaluation

4.1 Synthetic Dataset

In this experiment, the performance of the proposed method is evaluated with a synthetic network as in [14]. A graph stream including 16 random graphs are generated. In each graph snapshot, a total of 128 nodes is partitioned into 4 known equal-size communities of 32 nodes. Three synthetic change points separate the stream into 4 fragments: $G^{(1)}$–$G^{(4)}$, $G^{(5)}$–$G^{(8)}$, $G^{(9)}$–$G^{(12)}$ and $G^{(13)}$–$G^{(16)}$. The community structure is shown in Table 1.

Table 1. The community structure of the dynamic graphs

Graphs	Community 1	Community 2	Community 3	Community 4
$G^{(1)}$–$G^{(4)}$	{1–32}	{33–64}	{65–96}	{97–128}
$G^{(5)}$–$G^{(8)}$	{121–128}∪{1–24}	{25–56}	{57–88}	{89–120}
$G^{(9)}$–$G^{(12)}$	{113–128}∪{1–16}	{17–48}	{49–80}	{81–112}
$G^{(13)}$–$G^{(16)}$	{105–128}∪{1–8}	{9–40}	{41–72}	{73–104}

Suppose each node has on average 12 edges within the community and 4 edges to members of other communities. The weight of an edge is drawn uniformly from 1 to 10 for intra-community edges, while from 1 to 6 for inter-community edges.

The proposed change point detection approach is applied to the above synthetic network. Each graph segment is assumed to consist of one snapshot. The data is preprocessed by the Log-Entropy method. In the following, it is assumed

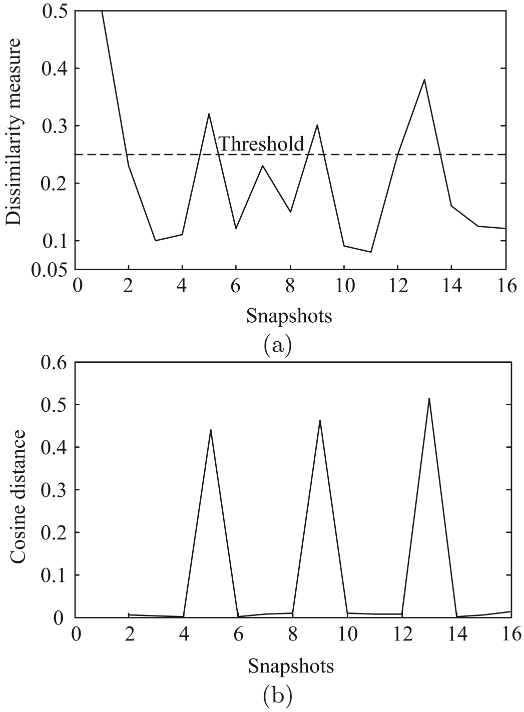


Fig. 1. Dissimilarity between consecutive snapshots: (a) dissimilarity measure computed by the method in [14]; (b) cosine distance calculated by the proposed method.

that $k = 3$, unless specified otherwise. The cosine distances between consecutive segments are depicted in Fig. 1(b). The cosine distance corresponding to point t on the x-axis is the dissimilarity between snapshots $t - 1$ and t. Figure 1(b) obviously shows that the proposed approach has detected all the three change points.

For comparison, Fig. 1(a) demonstrates the dissimilarity calculated by the method proposed in [14]. It is noticed that the dissimilarity measure therein cannot clearly determine the three change points. Moreover, the result strongly depends on the predefined threshold. In our result Fig. 1(b), fortunately, the cosine distance corresponding to a change point tremendously deviates from that for the stationary part. Therefore, the change points are more obviously identified by the proposed algorithm.

4.2 Enron Email Dataset

The Enron email dataset consists of the email communications in Enron Corp. from Jan 1999 to July 2002 [23]. A node represents an individual employee and an edge represents an email exchange between two employees. A most densely connected subgraph with 155 nodes is considered. Most of them correspond to

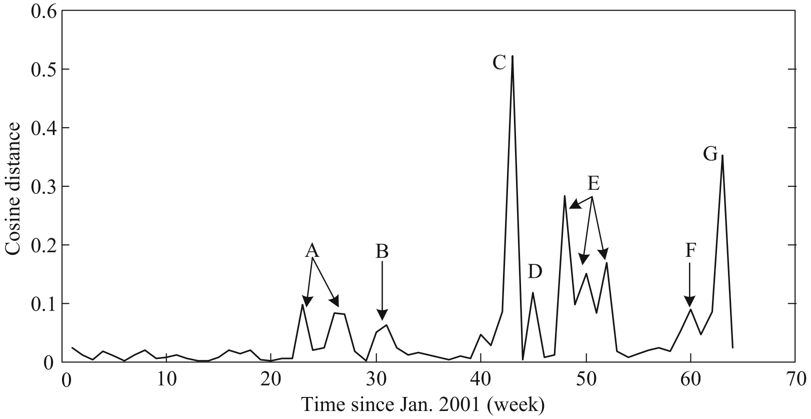


Fig. 2. Cosine distance between consecutive graph segments of the Enron dataset.

Table 2. Change points v.s. important events

Change points	Important events
A	Jun 2001: Rove divests his stocks in energy
B	Aug 2001: Kenneth Lay takes over as CEO
C	Oct 2001: Enron draws down $3 billion credit line
D	Nov 2001: Enron restates 3rd quarter earnings
E	Dec 2001: Enron is Bankrupted
F	Feb 2002: Lay implicated in plot to inflate profits and hide losses
G	Mar 2002: The indictment reported by the Wall Street Journal on January 28th is handed down

the core members in the executive committee. The directed sender-to-recipient graphs are constructed on a weekly basis. That is, the length of each graph segment is a week. We restrict to an interval of 65 weeks from Jan. 2001 to Mar. 2002. In each cumulative graph, the weight $w_{i,j}$ of a link represents the number of emails sent from employee i to j in that week. The Log-Entropy weighting is applied to the raw data.

Using the proposed edge-centric approach, the cosine distance between consecutive graph segments is illustrated in Fig. 2. As shown in the result, several change points are detected, which correspond to important events in Enron Corp., as listed in Table 2. Notice that the algorithm in [13] is also tested on the Enron dataset. However, as shown in [13], the algorithm therein only correctly detects the change point B. Clearly, experiments on real dataset also show that our proposed method effectively detects the change points in a dynamic network.

5 Conclusions and Future Work

This work proposes an edge-centric approach to detect the change points of the structure of dynamic networks. An edge-segment matrix is defined. The proposed approach uses an LSA based technique on the edge-segment matrix. Then each graph segment is represented by a low-dimensional vector. The dissimilarity between consecutive graph segments can be computed as the cosine distance between their representative vectors. A change point is declared if the cosine distance is relatively large. The proposed method is general enough in that it is applicable to all kinds of dynamic graphs (e.g., directed/undirected, weighted/unweighted ones). Evaluations on both synthetic and real datasets show that the proposed method effectively detects the change points in dynamic networks.

Future work. In the future, we will perform more experimental evaluation of the proposed method on different datasets. More detailed analysis on the choice of the dimensionality k will also be conducted.

Acknowledgments. This work was partly supported by the Tsinghua-Qualcomm Joint Research Program.

References

1. Rui, Z., Yanchao, Z., Jinyuan, S., Guanhua, Y.: Fine-grained private matching for proximity-based mobile social networking. In: Proceedings of IEEE Infocom, pp. 1969–1977 (2012)
2. Ide, T., Kashima, H.: Eigenspace-based anomaly detection in computer systems. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 440–449. ACM (2004)
3. Phua, C., Lee, V., Smith, K., Gayler, R.: A comprehensive survey of data mining-based fraud detection research (2010). arXiv preprint:1009.6119
4. Mutlu, A.Y., Aviyente, S.: Dynamic network summarization using convex optimization. In: Statistical Signal Processing Workshop (SSP), pp. 117–120. IEEE (2012)
5. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. Discourse Process. **25**(2–3), 259–284 (1998)
6. Tantipathananandh, C.: Detecting and Tracking Communities in Social Networks. Northwestern University, Evanston (2013)
7. Pietilanen, A., Diot, C.: Dissemination in opportunistic social networks: the role of temporal communities. In: Proceedings of the 13th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 165–174. ACM (2012)
8. Newman, M.E.: Modularity and community structure in networks. Proc. Nat. Acad. Sci. **103**(23), 8577–8582 (2006)
9. Folino, F., Pizzuti, C.: An evolutionary multiobjective approach for community discovery in dynamic networks (2013). http://www.computer.org/csdl/trans/tk/preprint/06573961-abs.html
10. Priebe, C.E., Conroy, J.M., Marchette, D.J., Park, Y.: Scan statistics on enron graphs. Comput. Math. Organ. Theor. **11**(3), 229–247 (2005)

11. Park, Y., Priebe, C., Youssef, A.: Anomaly detection in time series of graphs using fusion of graph invariants (2012). arXiv:1210.8429
12. Sun, J., Faloutsos, C., Papadimitriou, S., Yu, P.S.: Graphscope: parameter-free mining of large time-evolving graphs. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 687–696 (2007)
13. Koujaku, S., Kudo, M., Takigawa, I., Imai, H.: Structual change point detection for evolutional networks. In: Proceedings of the World Congress on Engineering (2013)
14. Duan, D., Li, Y., Jin, Y., Lu, Z.: Community mining on dynamic weighted directed graphs. In: Proceedings of the 1st ACM International Workshop on Complex Networks Meet Information and Knowledge Management, pp. 11–18 (2009)
15. Golub, G.H., Van Loan, C.F.: MatRix Computations, vol. 3. JHU Press, Baltimore (2012)
16. Dumais, S.T.: Improving the retrieval of information from external sources. Behav. Res. Methods Instrum. Comput. **23**(2), 229–236 (1991)
17. Landauer, T.K., McNamara, D.S., Dennis, S., Kintsch, W.: Handbook of latent semantic analysis. Psychology Press, Hove (2013)
18. Dumais, S.T.: Latent semantic analysis. Annu. Rev. Inform. Sci. **38**(1), 188–230 (2004)
19. Lehoucq, R.B., Sorensen, D.C.: Deflation techniques for an implicitly restarted Arnoldi iteration. SIAM J. Matrix Anal. Appl. **17**(4), 789–821 (1996)
20. Lanczos, C.: An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators. United States Government Press Office, Washington (1950)
21. Parlett, B.N.: The Symmetric Eigenvalue Problem, vol. 7. SIAM, Philadelphia (1980)
22. Zha, H., Simon, H.D.: On updating problems in latent semantic indexing. SIAM J. Sci. Comput. **21**(2), 782–791 (1999)
23. http://www.cs.cmu.edu/~enron/

Toward Automatic Censorship Detection in Microblogs

Donn Morrison(✉)

Department of Computer and Information Science,
Norwegian University of Science and Technology, Trondheim, Norway
donn.morrison@idi.ntnu.no

Abstract. Social media is an area where users often experience censorship through a variety of means such as the restriction of search terms or active and retroactive deletion of messages. In this paper we examine the feasibility of automatically detecting censorship of microblogs. We use a network growing model to simulate discussion over a microblog follow network and compare two censorship strategies to simulate varying levels of message deletion. Using topological features extracted from the resulting graphs, a classifier is trained to detect whether or not a given communication graph has been censored. The results show that censorship detection is feasible under empirically measured levels of message deletion. The proposed framework can enable automated censorship measurement and tracking, which, when combined with aggregated citizen reports of censorship, can allow users to make informed decisions about online communication habits.

1 Introduction

The recent and continuing popularity of the social aspect of the Internet, in particular social media and online social networks (OSNs), has facilitated unprecedented new ways of communication and levels of information sharing. This new freedom has challenged many governments, organisations and businesses which, for legitimate reasons or not, are struggling to control dissemination of news events, digital content and other sensitive information. When the censorship is acknowledged, justification ranges from maintaining public order and safety [9] to protection of morality from obscenity [20] to the protection of intellectual property or copyright [21]. In most cases, however, censorship is unquestionably a hindrance to a free and transparent society where citizens are able to participate by expressing ideas and opinions openly and without fear of reprisal.

Exposing censorship and the methods used to achieve it puts pressure on repressive elements and allows citizens to make informed decisions about how they participate in society. Projects such as Herdict [12,18] and ConceptDoppler [7] have undertaken to measure and track censorship online. However, Herdict relies on user reports and neither Herdict nor ConceptDoppler focus on user communication in OSNs.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 572–583, 2014.
DOI: 10.1007/978-3-319-13186-3_51

Recent research has focused on identifying sensitive keywords as well as influential or controversial users who are more likely to be censored [1,24,25]. However, these entities cannot always be anticipated before the censorship occurs, and therefore it is desirable to focus on features that are not content-based.

This paper proposes a novel method for censorship detection that does not rely on keyword lists or other forms of content. Instead, the approach classifies communication graphs derived from OSNs based solely on topological properties. More specifically, this work makes the assumption that user communication behaviour on microblogs, i.e., the posting and replying of messages, is generated by a random process that can be approximated using a graph generator. More importantly, however, is the further assumption that acts of censorship, specifically message deletion, results in a definite and measurable effect on the communication graphs.

Combined with the aforementioned citizen reports of censorship, OSNs and other online communities that deviate from known norms could be flagged as being censored and users could be warned to adapt strategies for organising and disseminating information.

This paper contributes to the understanding of the effects of social media censorship on network structure in the following two ways:

1. We identify salient topological features and show how they are affected by varying levels of censorship;
2. We propose a framework for automatic censorship detection at the network level that is content-agnostic.

In light of recent political events such as the Arab Spring of 2011, the current conflict in Syria and the ongoing censorship of media in China, there is an urgent need for a framework for the measurement of censorship online to ensure freedom of speech and access to information, and, in the larger context, maintain a free and open Internet. The approach aims to fill this gap and in doing so facilitate a better understanding of network censorship and ultimately provide a means for automated measurement, tracking and monitoring of censorship on the Internet.

The remainder of the paper is organised as follows. First we highlight related work that examines censorship of microblogs, primarily on the Sina Weibo network popular in China. Then in Sect. 3 we present the methodology which outlines the data generation, graph feature extraction and classifier setup. Finally, Sect. 4 presents the results and discussion.

2 Related Work

Censorship in the context of social media has been defined as the "suppression, limiting or deleting of objectionable" content or any other form of speech or expression [8,9]. There have been numerous works documenting instances and trends of censorship and circumvention strategies online, generally anecdotal or qualitative in nature, often relying on first-hand accounts [15,19,20]. However, some recent research has employed quantitative methods to measure and compare censorship practices on different OSNs [1,24,25].

Detection of deleted posts has been used in previous work to quantify censorship. The most pervasive methodology involves sampling microblog posts over a period of time to capture sensitive political events while querying the service at regular intervals to determine if any of the posts have been deleted [1,24,25].

Bamman et al. [1] uncovered politically sensitive terms more likely to be actively and retroactively deleted in a comparison between censorship on Twitter and China's Sina Weibo microblogging services. A random sample of collected messages found that 16.25 % were deleted from the Weibo network. Geographic distribution was found to have a strong impact on message deletion rates, with up to 53 % of sampled messages originating from some Chinese provinces deleted.

Initial research by [24] shows that active and retroactive censorship to a large extent succeeds in stemming the spread of information on microblogs. In a subsequent work, the authors studied the time distribution of deleted messages and found that nearly 30 % of deletions happen in the first 5–30 min and up to 90 % of deletions occur within 24 h of the posting [25]. Extrapolating the sampled data to message posting rates, the authors estimated that up to 4,200 workers working eight hour shifts would be required to match the demand for censorship levels on Sina Weibo alone. Furthermore, the authors uncovered censorship behaviour such as peak hours where censorship occurs and the practice of deleting entire repost cascades started from a single sensitive post. Ultimately, a complex array of censorship practices filter the continuous stream of Weibo posts such that sensitive topics do not enter into mainstream discussion.

Network perturbation and resilience is a closely related field where network metrics are studied under destructive processes that iteratively remove nodes or edges [4,11,23], however, these works do not consider censoring models for these processes nor do they formulate the problem as one of classification.

Despite these important works, no research to date has explored the effects of censorship on the underlying structure of the network and furthermore no research exists that attempts to automatically detect and classify censorship in these networks. Given that online social networks have certain universal properties [2], it is likely that common strategies of censorship such as limiting or deleting content or users from the network would have measurable effects on these properties. This research constitutes a first step to fill this gap by studying these effects.

3 Methodology

In this section we detail the methodology. First, we define a reply-graph over a microblog follow network. Then, we show how we use the *configuration model* to generate reply-graphs and present two methods to simulate censorship of these networks. Next, we introduce topological features extracted from the reply-graph that are then used to train a support vector machine in order to classify network censorship. Finally, our experimental setup is presented.

3.1 Definitions

Consider a directed multigraph $G = (V, E)$ without self loops where the nodes V represent users and the edges E represent microblog posts over the follow network. That is, an edge $e_{ij} \in E$ if user v_i is followed by user v_j *and* a post from v_i is shown in v_j's timeline. Note, an edge e_{ij} does not imply an edge e_{ji}. This notation corresponds to the flow of information over the edges E. As an example, the user v_i, who is followed by the set of users S, posts a new microblog entry m. Then, for each $v_j \in S$, a new edge e_{ij} is created in G, meaning that the entry m was visible in the timelines of users S.

To remain general, we refer to the graph G as the reply-graph as in [16], but with the constraint that an edge is only possible if there is a follow relationship between two users.

3.2 Configuration Model

Due to the limited availability of censored microblog reply-graphs, we have chosen to generate random graphs with similar characteristics. Simulation of network data is commonly used when access to data is limited or when characteristics of the network must be carefully controlled. Since our aim is the study of reply-graphs, we make use of the directed multigraph *configuration model* (CM) proposed by [17] that permits random graph construction with arbitrary in and out degree distributions.

Power laws have been observed in the degree distributions of online social networks [2] although the ubiquity of data conforming to this distribution is often overstated [3] and depending on the network in question a closer fit may be found in any number of exponential distributions (e.g., Pareto-lognormal distribution [10]). However, to simplify network generation in this preliminary work we assume the degree distributions follow a power law and generate the reply-graphs accordingly. We fix the power law exponent to $\alpha = 2.0$ for both the in and out degree distributions that are used as input to the CM and set the network size to $|V| = 1000$ nodes.

3.3 Simulating Censorship

We focus on the censorship of microblog posts which are represented by the edges E of G. That is, we do not consider the case where user accounts (the nodes V of G) are suspended or deleted. Two censorship strategies are compared. The first is based on a uniform sampling of a fraction of the edges in G. This strategy can be likened to a population of users that are subjected to uniform censorship, that is, each user's post has the same probability of being deleted. This carries with it the assumption that each user is equally likely to post about a topic considered worthy of censorship, which is unlikely to be the case in the real world [25]. Nonetheless, a uniform sampling of deleted edges serves as a useful baseline. The second strategy is based on the removal of *repost* (or *retweet*) cascades. Such a cascade occurs when users, upon seeing a message posted by one of their followees, choose to repost the message to their followers. The information cascades over the network according to the popularity of the original post. Censoring of entire cascades has been empirically documented in previous work [25] where censors were shown to retroactively remove a post and all subsequent reposts. We simulate repost cascades using the independent cascade model (ICM).

ICM, originally proposed by Kempe *et al.* [13], is used to model spreading processes such as disease or information cascading through OSNs. The method starts with a set of activated (infected) seed nodes and at each iteration a coin is flipped to determine whether the information flows across an outgoing edge of a newly activated node. On success, the information spreads and the activated node attempts to spread the information in the next iteration. A global transmission probability is used for the coin flip and in our experiments this probability is set to 0.1 to allow for larger cascades that will form the censored edges of G. For simplicity, the seeds are selected by ranking the nodes according to out degree and taking the top five, corresponding to the fraction 0.005 of the nodes V in G. Cascades are generated with ICM until the total number of edges reaches the censorship threshold. Figure 1 shows an example set of cascades generated over G using ICM.

For both the uniform and ICM-based censorship strategies, we remove a fraction γ of the edges in G for each $\gamma \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$ where the total number of edges removed equals $\gamma \times |E|$. This range allows us to study the detection of levels of censorship that have been empirically measured [1].

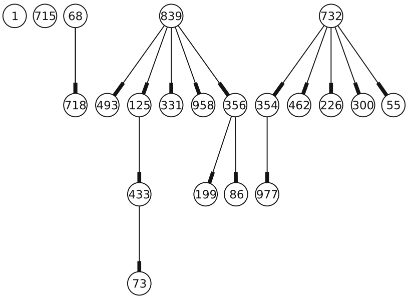


Fig. 1. An example set of generated repost cascades starting with three initially activated seed nodes (top row).

3.4 Network Features

Communication graphs derived from OSNs have specific characteristics distinguishing them from other networks such as random graphs [2,10,14,22]. For example, node degree distributions of OSNs have been shown to be exponentially distributed, following power laws [2] or Pareto-lognormal mixtures [10]. OSNs also exhibit small diameters [2] that shrink with network growth [14] and have a higher number of triangles[1] compared to random graphs [2]. Centrality measures also show characteristic behaviour. In [6] the authors measure the stability of various centrality measures of OSNs sampled at different thresholds and show that certain centrality measures are less robust to uniform sampling. Other research has examined spectral eigenvalue distributions for classification of biological networks as the eigenvalues are known to summarise various topological properties of graphs [22].

We motivate our choice of graph features by drawing on the aforementioned work. Based on these characteristics, we derive the following features and posit that they can be used to discriminate between censored and uncensored reply-graphs based on the assumption that acts of censorship fundamentally change network structure. We abbreviate feature names in parentheses for use in the figures and tables that follow.

Average degree (avgdeg) is defined as $\frac{2\times|E|}{|V|}$, ignoring edge direction. The **assortativity coefficient** (assort) is the degree correlation between pairs of connected nodes for the undirected equivalent of G. The **diameter** (dia) of G is defined as the maximum shortest path for the undirected equivalent of G and the **radius** (rad) is the minimum of the set of maximum path lengths from every node to every other node in the undirected equivalent of G.[2] The **average clustering coefficient** (clustering) is defined as $\frac{1}{|V|}\sum_{i\in V} C_i$ where C_i is the local clustering coefficient measuring the number of edges divided by the number of total possible edges between the neighbours of node i for the undirected equivalent of G. The **average betweenness centrality** (betcent) is the average of the number of shortest paths that pass through any node in G.

For simplicity, we assume the in and out degree distributions of G to follow a power law. As such, we include the estimates of the **power law exponent** α (in_alpha_fit and out_alpha_fit, respectively) as well as the goodness of fit measured by the **negative log-likelihood** (in_likelihood_fit and out_likelihood_fit). The parameters are estimated by maximum likelihood estimation (MLE) as described in [3]. Finally, we calculate and retain the first 50 **eigenvalues of the Laplacian matrix** (spec0-49).

The resulting feature vector F is of length 60 (10 topological features plus 50 Laplacian eigenvalues).

3.5 Classification

The classifier used in this work is the support vector machine (SVM) [5] with the radial basis function (RBF) as a kernel with parameters complexity $C = 1.0$ and gamma $g = 0.01$. The choice of classifier and kernel is motivated by satisfactory experimental results and the pervasive use of SVMs in machine learning literature although we note that any number of classification methods could be

[1] If actors A and B are connected and B and C are connected, there is a high probability that actors A and C are also connected.

[2] Diameter and radius are calculated on the largest connected component.

readily used. For brevity we omit details of SVMs and statistical learning theory and refer the reader to [5].

3.6 Experimental Setup

The experimental setup is as follows:

1. Generate $N = 100$ directed multigraphs G with $|V| = 1000$ nodes using the CM
2. Simulate censorship uniformly and with ICM by removing $\gamma \times |E|$ edges, yielding G^{γ}_{cu} and G^{γ}_{cICM}
3. Compute topological features F, F^{γ}_{cu} and F^{γ}_{cICM} of G, G^{γ}_{cu} and G^{γ}_{cICM}, respectively
4. Classification by pairwise 10-fold cross validation on (F, F^{γ}_{cu}), (F, F^{γ}_{cICM}) with class labels $\{0, \gamma\}$

To account for variance, Step 4 is repeated 10 times, each using a different random seed. The Java-based WEKA machine learning toolkit is used for classification and feature selection[3] and all experiments are conducted on an Intel quad-core i5-2520M CPU laptop running at 2.50 GHz.

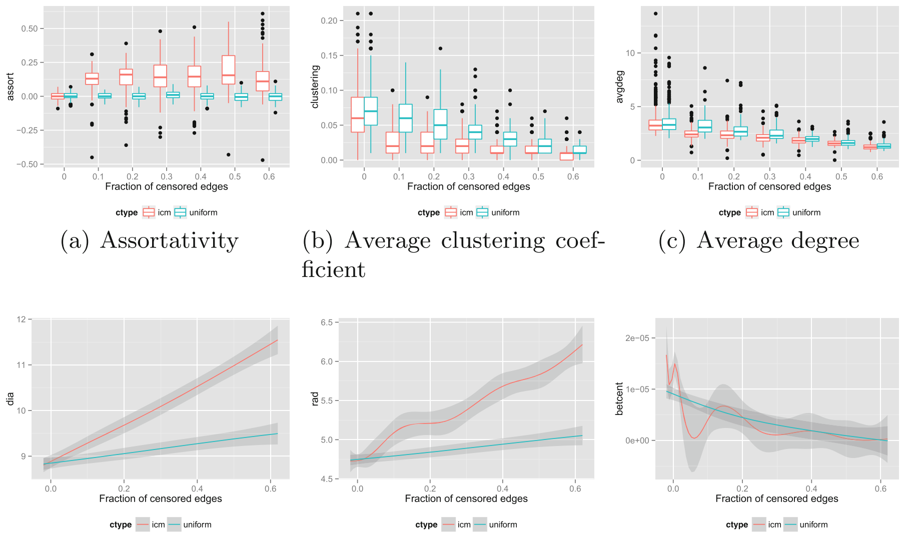
4 Results and Discussion

In this section we (1) show the effects of censorship on graph features, (2) present the classification results and (3) highlight salient graph features discovered through feature selection. Some figures have been fit with a statistical smoother and include shaded 95 % confidence intervals for readability.

4.1 Censorship Effects on Graph Features

Figure 2 shows the effect of censorship on the topological network features. In (a), we see that under uniform censorship, assortativity is not a discriminating feature. However, for censorship of repost cascades with ICM, we see a substantial increase in assortative nodes for mid to moderate levels of edge removal. At $\gamma = 0.6$, there are less assortative nodes since at this high level much of the network structure has been lost. The average clustering coefficient (b) shows different trends for uniform and ICM. For uniform it peaks at $\gamma = 0.1$ and then slowly declines while for ICM the decline is immediate with a fluctuating mean for larger values of γ. The trends for average degree (c) are similar to both strategies with a predictable decline as γ increases. Unsurprisingly, network diameter (d) and radius (e) behave similarly because they both describe characteristics of the shortest paths. For ICM, the metrics increase with γ, consistent with previous work [14]. For uniform edge removal, the growth is slower. Average betweenness centrality (f) shows a declining behaviour as γ increases.

[3] http://www.cs.waikato.ac.nz/ml/weka/

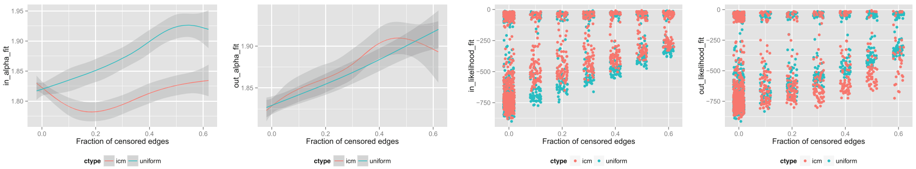


(a) Assortativity (b) Average clustering coefficient (c) Average degree

(d) Diameter (e) Radius (f) Average betweenness centrality

Fig. 2. Topological features as a function of the fraction of censored edges.

Moving our focus to the in and out degree distributions, Fig. 3(a) and (b) show the estimated exponent α of the power law for the in and out degree distributions, respectively. Recall that the networks were generated with a value of $\alpha = 2.0$. The estimated value of α is lower than expected, possibly due to the scale of the network and the fact that the configuration model may not accurately portray the given distributions in the resulting network. However, the estimations vary with γ, which indicates discrimination potential. The likelihoods of the power law fitting, shown in (c) and (d), are also informative features. As we deviate from the power law in the uncensored network by simulating censorship, the distributional fit becomes less and less accurate. While the power law



(a) Estimated in degree α (b) Estimated out degree α (c) In degree α likelihood (d) Out degree α likelihood

Fig. 3. Estimated power law exponent α and log-likelihood values for in and out degree distributions as a function of the fraction of censored edges.

assumption is simplistic, we expect that this can be readily generalised to other distributions if the degree sequences for uncensored networks are known a priori.

For satisfactory pairwise classification of censored versus uncensored networks, discriminating features should exhibit different values for $\gamma = 0.0$ and $\gamma >= 0.1$. Through visual inspection of Figs. 2 and 3,[4] we can see that when used together, the chosen features appear to support our hypothesis that censorship, even at lower levels ($\gamma = 0.1$), fundamentally alters network structure.

4.2 Classifying Censorship

Figure 4 shows the classification accuracy of the SVM as a function of varying the fraction of censored edges for the uniform and ICM-based censorship strategies. Accuracy is substantially higher than random (50 %), and quite satisfactory for a two-class problem. We see that censorship of repost cascades (ICM) has a much stronger affect on network structure for lower values of γ than the uniform strategy due to the inherent structural correlation between repost cascades and the reply-graph. Interestingly the accuracy plateaus to 97 % at $\gamma = 0.6$ which indicates that censorship at $\gamma >= 0.5$ is trivial to detect. For uniform edge removal, there is a steep transition between $\gamma = 0.2$ and $\gamma = 0.4$ after which it matches classification accuracy of ICM.

Table 1. Feature selection results for the two censorship strategies.

Censorship strategy	γ	Selected attributes
ICM	0.1	assort, spec2, spec3, spec38
	0.2	avgdeg, in_alpha_fit, out_alpha_fit, assort, spec21
	0.3	avgdeg, in_alpha_fit, assort, rad, betcent, spec1, spec13, spec29, spec35, spec48
	0.4	out_alpha_fit, assort, rad, spec1, spec12
	0.5	out_alpha_fit, assort, rad, spec0
	0.6	out_alpha_fit, assort, rad, spec2
Uniform	0.1	in_alpha_fit, spec1, spec10, spec13
	0.2	clustering, betcent, spec22, spec29, spec36
	0.3	dia, rad, clustering, betcent, spec12, spec13, spec15
	0.4	avgdeg, dia, rad, spec1, spec7, spec17, spec19, spec30, spec34, spec35
	0.5	avgdeg, dia, rad, clustering, betcent, spec0, spec5, spec9, spec18, spec28, spec32, spec45, spec47
	0.6	clustering, spec8

[4] We omit plots for the Laplacian eigenvalues due to space considerations.

4.3 Feature Selection

Feature selection was performed for each γ using a greedy forward search on the entire dataset with the RBF SVM for both uniform and ICM. These results are presented in Table 1. For ICM, topological features such as assortativity and radius appear to be selected for most values of γ along with the in and out degree α estimation and various spectral eigenvalues. For uniform edge removal, average degree, clustering, diameter and radius are selected as well as betweenness centrality and some spectral eigenvalues. The MLE estimation of α is mostly absent, possibly due to high correlation with other features.

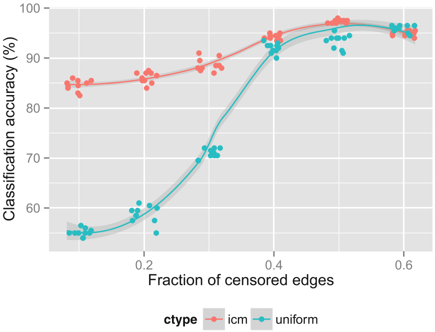


Fig. 4. Classification accuracy as a function of the fraction of censored edges.

5 Conclusion

In the cat and mouse game of censorship and circumvention, sensitive word lists play a central role and are invaluable for measuring censorship [1]. However, in this paper we have shown that network structure is also a very promising avenue for measurement and detection of censorship. We examined the feasibility of automatically classifying networks as either censored or uncensored based on topological features. We compared two censorship strategies: (1) a uniform strategy where every post has an equal probability of being removed and (2) a strategy based on removing entire repost cascades. As expected, deletion of repost cascades was shown to result in higher classification accuracy. In the real world, however, models of censorship are far more complex and involve sensitive topics, users, as well as a combination of seemingly arbitrary post removals.

We identified salient topological properties including assortativity, average degree, deviations from scale-free degree distributions and average clustering coefficient that provide a starting point for exploring other local and global network features in the context of censorship detection.

There are some shortcomings in the present work. First, both the power law assumption and the configuration model for network generation are simplistic, so other degree distributions and network generators need to be examined. Second, we ignored the problem of sampling an online social network by directly generating the communication graphs. In reality, it not feasible to collect the complete communication graph due to the scale of the data. Thus, a future work will incorporate network sampling into the methodology to show how this affects classification. This is expected to negatively impact classifier accuracy. Third, the scale of the simulated networks size is small, with $|V| = 1000$, however, we expect that for larger networks the features and subsequent classification results will stabilise, although at the cost of increased complexity. Finally, the methods presented in this preliminary study must be validated on real data. For this to be feasible it may be necessary to use different online social networks as sources of censored and uncensored reply-graphs.

There are several directions in which this work can be extended. Given that censorship primarily affects the diffusion of information, in addition to edge removal, we will examine how different levels of node censorship (i.e., suppression or removal of user accounts) affects the spread of information through an online social network. Second, the classification framework could be extended to provide a quantitative estimation of the level of censorship (i.e., the estimation of γ) in a given online social network.

Acknowledgments. This work was funded by the European Union (EU) and Science Foundation Ireland (SFI) in the course of the projects ROBUST (EU grant no. 257859) and CLIQUE Strategic Research Cluster (SFI grant no. 08/SRC/I1407), and LION-2 (SFI grant no. SFI/08/CE/I1380).

References

1. Bamman, D., O'Connor, B., Smith, N.A.: Censorship and deletion practices in Chinese social media. First Monday **17**(3) (2012)
2. Chakrabarti, D.: Graph mining: laws, generators, and algorithms. ACM Comput. Surv. (CSUR) **38** (2006). http://dl.acm.org/citation.cfm?id=1132954
3. Clauset, A., Shalizi, C.R., Newman, M.E.J.: Power-law distributions in empirical data. SIAM Rev. **51**(4), 661–703 (2009)
4. Cohen, R., Erez, K., Ben-Avraham, D., Havlin, S.: Breakdown of the internet under intentional attack. Phys. Rev. Lett. **86**(16), 3682–3685 (2001)
5. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
6. Costenbader, E., Valente, T.W.: The stability of centrality measures when networks are sampled. Soc. Netw. **25**(4), 283–307 (2003)
7. Crandall, J.R., Zinn, D., Byrd, M., Barr, E.T., East, R.: ConceptDoppler: a weather tracker for internet censorship. In: ACM Conference on Computer and Communications Security, pp. 352–365 (2007)
8. Deibert, R.: Black code redux: censorship, surveillance, and the militarization of cyberspace. In: Boler, M. (ed.) Digital Media and Democracy: Tactics in Hard Times, pp. 137–164. MIT Press, Cambridge (2008)

9. Dick, A., Oyieke, L., Bothma, T.: Are established democracies less vulnerable to Internet censorship than authoritarian regimes?: The social media test. Technical report, Committee on Freedom of Access to Information and Freedom of Expression (FAIFE), University of Pretoria, South Africa (2012)
10. Fang, Z., Wang, J., Liu, B., Gong, W.: Double pareto lognormal distributions in complex networks. In: Thai, M.T., Pardalos, P.M. (eds.) Handbook of Optimization in Complex Networks, pp. 55–80. Springer, New York (2012)
11. Gallos, L.K., Argyrakis, P., Bunde, A., Cohen, R., Havlin, S.: Tolerance of scale-free networks: from friendly to intentional attack strategies. Phys. A: Stat. Mech. Appl. **344**(3), 504–509 (2004)
12. Hwang, T.: Herdict: a distributed model for threats online. Netw. Secur. **2007**(8), 15–18 (2007). http://www.sciencedirect.com/science/article/pii/S1353485807700740
13. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network, p. 137 (2003)
14. Leskovec, J., Kleinberg, J.M., Faloutsos, C.: Graph evolution: densification and shrinking diameters. TKDD **1**(1), 1–40 (2007)
15. MacKinnon, R.: Consent of the Networked: The Worldwide Struggle For Internet Freedom. Basic Books, New York (2012)
16. Morrison, D., Mcloughlin, I., Hogan, A., Hayes, C.: Evolutionary clustering and analysis of user behaviour in online forums. In: Proceedings of ICWSM-12, 6th International AAAI Conference on Weblogs and Social Media (2012)
17. Newman, M.E., Strogatz, S.H., Watts, D.J.: Random graphs with arbitrary degree distributions and their applications. Phys. Rev. E **64**(2), 026118 (2001)
18. Nuss, J.: Web site tracks world online censorship reports, 4 August 2009. http://phys.org/news168613309.html
19. Roberts, H., Zuckerman, E., Palfrey, J.: Circumvention tool evaluation. Technical report (2011)
20. Sang-Hun, C.: Korea Policing the Net. Twist? Its South Korea, 12 August 2012. http://www.nytimes.com/2012/08/13/world/asia/critics-see-south-korea-internet-curbs-as-censorship.html
21. Stone, B.: The Inexact Science Behind D.M.C.A. Takedown Notices, 5 June 2008. http://bits.blogs.nytimes.com/2008/06/05/the-inexact-science-behind-dmca-takedown-notices/
22. Takahashi, D.Y., Sato, J.R., Ferreira, C.E., Fujita, A.: Discriminating different classes of biological networks by analyzing the graphs spectra distribution. CoRR abs/1208.2976 (2012)
23. Yadav, G., Babu, S.: NEXCADE: perturbation analysis for complex networks. PloS One **7**(8), e41827 (2012)
24. Zhu, T., Phipps, D., Pridgen, A., Crandall, J.R., Wallach, D.S.: Tracking and quantifying censorship on a chinese microblogging site. CoRR abs/1211.6166 (2012)
25. Zhu, T., Phipps, D., Pridgen, A., Crandall, J.R., Wallach, D.S.: The velocity of censorship: High-fidelity detection of microblog post deletions. CoRR abs/1303.0597 (2013)

Competition Component Identification on Twitter

Cheng-Huang Yang, Ji-De Chen, and Hung-Yu Kao(✉)

Department of Computer Science and Information Engineering,
National Cheng Kung University, Tainan, Taiwan, R.O.C.
chyang@ikmlab.csie.ncku.edu.tw,
twooo333chen@gmail.com, hykao@mail.ncku.edu.tw

Abstract. Twitter becomes a popular microblogging platform that let people to express their opinions on the web in recent years. Companies with new products always want to find consumers or public opinions about their products and services after they released new products. Due to this reason, more and more researchers use the opinion mining technology on this microblog media that contains abundant and real-time information, to extract useful opinions and information. In this paper, we aim to mine the opinions on Twitter and further extract the competition relations discussed on Twitter. For Example, if we want to know how people express their opinion about "Packer" (an American football team name), we also want to know what the Packer's competitors are. In this paper, we introduce a hashtag graph and use the ranks in this graph to represent the competition behavior and competition components (competitors).

Keywords: Twitter · Microblog · Opinion mining · Sentiment analysis

1 Introduction

In recent, after the big growth of blogs, microblogging service like Twitter, Plurk and Google+ has received more and more attention. A microblog differs from a traditional blog in that its content is typically smaller in both actual and aggregate file size [1]. These small message make users can easily share their daily life, express opinion on something and obtain instant information.

Twitter, one of the most popular microblogging services allows user (twitterer) to post message (tweet) through the website interface, SMS, or a range of apps for mobile devices. It has over 140 million active users, generation over 340 million messages per day [2]. As a result of rapidly increasing numbers of tweets, Twitter becomes valuable source for mining people's opinion and sentiment. In addition that easy accessibility of Twitter can not only enable information sharing in real-time but also allows companies to quickly know that the opinion expressed by customer about their products after they are released.

Businesses always want to find consumer or public opinions about their products and services after they released. Intuitively, a newly released product that causes a lot of online discussions is a way to give expression to the experience of users. User can express their opinion about product or service through review site, forum discussion,

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 584–595, 2014.
DOI: 10.1007/978-3-319-13186-3_52

blogs, and social networks, and we call them "review" in the rest of the paper. Many of pervious works use text mining technology to mine the review on the web to realize how the opinion expressed about some product. This area of study is called "opinion mining" or "sentiment analysis". And this technology can replace the opinion polls, surveys, and focus groups which are used to gather public opinions about its products.

Tradition opinion mining from review on the web focuses on specific domain such as movie, book or consumer product. Figure 1 shows that the Liu et al. [3] proposed a system with a visual comparison of consumer opinions of different. The authors find out which feature has opinion expressed on it, and identify the opinion polarity regarding those features. Each bar in Fig. 1 shows the percent of reviews that express positive (above x-axis) and negative (below x-axis) opinions on a feature of a camera. We can easily see that digital camera 1 is a superior camera.

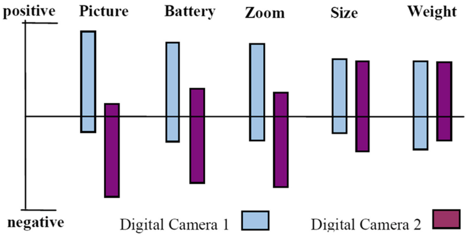


Fig. 1. The opinion comparison shown in [3]

With the explosive growth of microblog, opinion mining application on it also received more and more attention in resent year. O'Connor et al. [4] link the sentiment on Twitter to public polls. Chamlertwat et al. [5] analysis the tweets about smartphone and extract the features and its opinion.

In our work, we change the view on opinion mining question. If opinion polarity shows that product 1 performs better than product 2 on the feature. It means that product 1 is indeed better than product 2 or product 2 worse than product 1 on this feature. Like the above case, camera 1 is better than camera 2 and camera 2 worse than camera 1 on "battery" feature both lead opinion polarity shows that camera 1 performs better than camera 2. In other word, opinion polarity or sentiment polarity is relative. Because of opinion polarity or sentiment polarity is relative, we want to know can camera 2 to be found when we mining the opinion about camera 1. We define the competition component like camera 1 and camera 2 that both components contain relative opinion polarity between them.

About competition, Wikipedia give it a definition:

> "Competition in biology, ecology and sociology is a contest between organisms, animals, individuals, groups, etc. for territory, a niche, or a location of resources, for resources and goods, for prestige, recognition and awards, for mates and group or social status, for leadership." [6]

As Wikipedia says, the competition occurs if and only if two or more components compete over same or similar resources. In real world, hTC, Samsung and Nokia are all smartphone manufacturers, and they release various smartphones to grab more market share. In other words, hTC, Samsung and Nokia are components, compete over smartphone market share. Customer will consider the price, performance, even the build-in software as features when they are choosing a smartphone. McCain compete with Obama, two presidential candidates, both of them want to win presidential election in 2008. In this case, McCain and Obama (components) all need to get vote (resources) as much as they can in election. Voter will take candidates' political party, political views or their personality into account when they give their vote to one of the presidential candidates. Twitter provide us a real-time information platform, we can mine the latest opinion on it. Identifying the competition component on Twitter allow us know which component is competitor at this time.

Although identify the competition component on Twitter is similar to tradition opinion mining question, there are something different between them.

First, not like tradition way, those always focus on specific domain with explicit component to assign opinion polarity. For the sake of finding which component may have relative opinion polarity is an important step in identifying the competition component.

Secondly, in order to identify competition component, we need to find out the feature about the component. Furthermore, only the same features can be put together and compare it's opinion. It is common that people use different words or phrases to describe the same feature, for instance, “photograph”, “photo”, “pic”, “picture” and “image” all refers to the same feature in digital camera review. It is necessary to find a way to build a feature synonym set. Since feature synonym set be built, the features can match up together if they express in different words. However feature synonym and matching problem is another problem need to solve in our work.

As we mention above, tradition opinion mining always focus on specific domain to extract feature. Like Hu and Liu [7, 8], they solved synonym and feature matching problem easily by using WordNet [9] to build the feature synonym set. Unfortunately, there is no easy way to solve this problem in our work. To take a simple example, we assume two components are “Google” and “Apple” respectively. “Android” and “iOS” are the features extract from Google and Apple. The “Android” and the “iOS” all refer to mobile operating system distributed by Google and Apple respectively. But we cannot find the semantic relation between them through WordNet. Therefore, we need an alternative to solve this problem.

2 Related Works

In the approach proposed by Hu and Liu [7, 8], there were two types of the product features: frequent features and infrequent features. The frequent feature indicates the features that are usually mentioned in user reviews, and the uncommon features are the infrequent features. The authors observed that the features of a product almost formed by the noun, although user writes reviews in different phraseology. Based on the observations, they use data mining system to extract the frequent noun items as frequent feature candidates. The frequent features usually accompany with opinion words. If there is any opinion word not occurs with frequent features, they try to extract the noun or noun phrase which is not frequent feature and appear nearest opinion words in a sentence, and define the noun or noun phrase as infrequent feature.

Liu et al. [3] used a supervised rule discovery to extract the product features. First, they generate Part-of-Speech tag for each word. After tagging, the product features are picked manually. The product features are given a label [feature], and they utilize association mining system CBA to look for rules of occurrence of [feature]. The rules which they generated can be applied to find out the feature they really want in reviews.

Turney [10] used semantic orientation (SO) to assign the sentiment of the words which extract from reviews. Author observed that the word of positive semantic orientation would appear together with positive reference opinion word more than with negative reference opinion word. Otherwise, word of negative semantic orientation appear closed to negative reference opinion word more than with positive reference opinion word.

Based on this observation, Turney presented a Semantic Orientation (SO) mining method based on Pointwise Mutual Information (PMI) for sentiment assignment. PMI is a measurement usually used to calculate the correlation degree between the two words. And PMI is defined as follows:

$$PMI(word_1, word_2) = log_2\left[\frac{p(word_1 \ \& \ word_2)}{p(word_1)p(word_2)}\right]$$

$p(\mathrm{word}_1 \ \& \ \mathrm{word}_2)$ is the probability that word_1 and word_2 appear together. $p(\mathrm{word}_1)$ and $p(\mathrm{word}_2)$ are the probabilities that word_1 and word_2 occur respectively. The SO of a phrase is calculated here as follows:

$$SO(phrase) = PMI(phrase, \text{"excellent"}) - PMI(phrase, \text{"poor"})$$

The phrase to be estimated would be identified as positive if the SO value was calculated more than zero, and negative otherwise.

Pang et al. [11] experimented several supervised machine learning techniques to classify the sentiment at the document level. Those approaches classify the document into classes, positive and negative. In order to train the classifier, authors selected some common text features like n-gram and part-of-speech tags in hand-labeled training data.

3 Method

3.1 Problem Definition

We describe the competition component problem more specifically as follows:

- *C*: the set of component
- *F:* the set of component's feature
- *O:* the set of feature's opinion polarity score [+1, −1]

One component $c_i \in C$ contains features and features' opinion:

$$c_i = \{F_i, O_i\}$$

And $o_{i_j} \in O_i$ is the opinion of the feature $f_{i_j} \in F_i$. The components are always discussed under a certain feature set. If two components have relative opinion between them on a feature, it means than they compete over this feature. We define this "component-feature-component" pair as a "competition behavior".

Two components c_1 and c_2 have relative opinion between them $\left(\left|o_{1_x} - o_{2_y}\right|\right)$, on the feature f_{1_x} and f_{2_y}. f_{1_x} and f_{2_y} are same feature or similar feature. Therefore "$c_1 - f_{1_x} - f_{2_y} - c_2$" composed a competition behavior.

We quantify composed a competition behavior into a numerical score called "CBS".

$$\mathrm{CBS}\left(c_1, c_2, f_{1_x}, f_{2_y}\right) = \left|o_{1_x} - o_{2_y}\right| * FMS(f_{1_x}, f_{2_y}),$$
$$\text{where } \left|o_{1_x} - o_{2_y}\right| > \alpha \ (\alpha \text{ is a threshold})$$

$FMS(f_{1_x}, f_{2_y})$ is a feature matching score. When two components' features $f_{1_x} = f_{2_y}$, and the $FMS(f_{1_x}, f_{2_y}) = 1$. It means that when two components' features completely similar to each other, CBS obtains all $\left|o_{1_x} - o_{2_y}\right|$ which called "relative opinion score". We can aggregate the CBS score after we have measured all features' CBS scores between components. A high CBS score indicates that two components compete with each other.

3.2 Method Flowchart

We briefly describe our method flowchart in this subsection, and we illustrate it on Fig. 2. After process the raw Twitter data, we extract the competition component candidates and component's feature at first. In order to extract competition behavior,

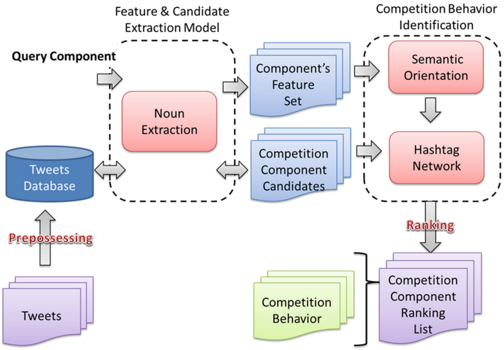


Fig. 2. Method flowchart

we use semantic orientation and a hashtag network to build a competition behavior identification model. At the last step, we aggregate all the information to find the competition component list.

3.3 Dataset

Our dataset is obtained from TREC 2011 microblog track. This dataset contains 15,576,810 tweets from 2011/01/23 to 2011/02/08. After we filtered the non-English tweets, the number of tweets in the left dataset is reduced to 4,084,579.

3.4 Data Preprocessing

Part-of-Speech Tagging (POS). According to our observing, competition component candidates and component features are usually nouns or noun phrases in tweets, and the words contain opinion are adjectives that appear near by the features. Thus the Part-Of-Speech (POS) tagging is crucial step. We use the POS tagger which reserved tagging tweets and it developed by Gimpel et al. [12]. For example, (N) indicates common noun and (A) indicates adjective. Because of this POS tagger cannot tag noun phrases; we use four easy rules to identify bigram noun phrase. These rules were inspired by the observation. We found that bigram noun phrase appear very often in tweets, and it is the combination of two types of word: common noun and proper noun. Table 1 describes the four rules and example.

Table 1. Noun PHRASE EXAMPLE

Rule	Example
(N) + (N)	video game
(N) + (^)	president Mubarak
(^) + (N)	Google map
(^) + (^)	Michael Jackson

Stemming and Remove the Stopwords. Although, after doing the word segmentation and POS tagging process, the data were still noisy. First, every word needs to pass the stemming process. Stemming is the process for reducing inflected words to their stem. For example, stemming reduces the words “stemmer”, “stemming”, and “stemmed” to the root word, “stem”.

In general, the words like “a”, “the” and “this” are treated as noisy word, because they appear too much times. We here apply the inverse tweet frequency to help us to identify the stopwords in our data.

The word *i*‘s inverse tweet frequency is defined as follow:

$$itf(i, T) = \log \frac{|T|}{|\{t \in T \ : \ i \in t\}|}$$

Here, $|T|$ denotes the total number of tweets and T is the set of all tweets, and $|\{t \in T : i \in t\}|$ is the number of tweets which contain the word i. itf is a measure of whether the term is common or rare across all tweets. Then we remove about three hundred words and fifty hashtags.

3.5 Feature and Candidate Extraction

Here, we need to find the competition component candidates for the query component c_q. We observed that competition components co-occur in one tweets which contain conjunction (like "and, or, &") or preposition (with, for, to).

We extract nouns and noun phrases as competition component candidates from tweets which contain c_q and conjunction term. It can simply construct a set of nouns and noun phrases contain competition component candidates. Next, PMI-IR can be used to reduce the competition component candidate set. PMI-IR is defined as follow:

$$PMI{-}IR(c_q, candidate) = log_2\left(\frac{frequence(c_q \ \& \ candidate)}{frequence\left(c_q\right) * frequence(candidate)}\right)$$

Where $frequence(x)$ denotes the number of tweets contain term, $frequence(x \ \& \ y)$ means the number of tweets contain term x and term y. PMI-IR is a measure for measuring the independence of two distinct terms. Information will be greater for two terms which are strongly dependent upon each other. We discard the feature which has high PMI-IR value and low PMI-IR value.

We also find that the words contain opinion are adjectives and appear near by the features. Hence, our system extracts nouns and noun phrases as component's feature from tweets which contain c_q and adjectives.

3.6 Feature Matching Between Components

Hashtag network: Hashtags in Twitter are the words or phrases prefixed with the symbol "#" such as "#hashtag". Twitterer use hashtag to categorize tweets and highlight topics. Table 2 shows four tweets which contain hashtags.

Table 2. Tweet with Multi Hashtags

Tweet
Should Google Increase its 20 % Time? http://bit.ly/hq6xXc **#socialmedia #tech**
Apple hires former NSA, Navy analyst as security czar http://twlv.net/rvikQD **#socialmedia #tech #apple**
#electronics #deals #sale #gadgets #technology #tech: Nokia plans to sack half its board: report http://bit.ly/gYG5Lk
New: Canon S35 Ink Cartridge for Canon Printers - Black **#tech #gadgets #electronics** http://bit.ly/h648u7

Above four tweets all discuss about different thing, but they all have same hashtag: "#tech" (i.e. technology). In this case, we observed important characteristic about the hashtags:

- Four tweets all relate to the topic "technology".
- Google, Apple, NSA, Nokia and Canon may have some semantic relation, because they both appear in the tweets about "technology".
- Electronic and technology have semantic relation because hashtag "#electronic" and "#tech" categorize together.

From mentioned the observation above, we build a hashtag semantic relation network for Twitter. On this network, every node is a hashtag. If two hashtags co-occur in one tweet, they will have an edge between them. To put it differently, nodes have some semantic relation between them if they have the path can link each other. The shorter path indicates the stronger semantic relation, and the longer path indicates the weaker semantic relation.

For example, if we only use the first tweet and the last tweet form Table 2 to build the hashtag network, the conducted network is shown in the Fig. 3.

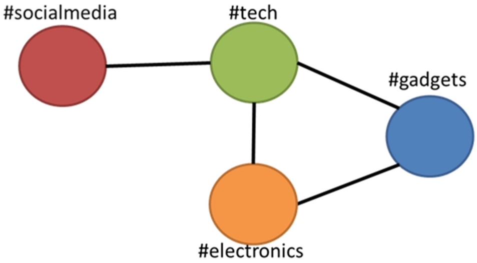


Fig. 3. An example of a hashtag network

Feature Matching. We show how hashtag network help us to matching the component's feature in Fig. 4. At the first step, we extract the hashtags that appear with the features. Like we mention above, hashtags can link to each other means that two hashtag have some semantic relation. We use this idea to matching the features.

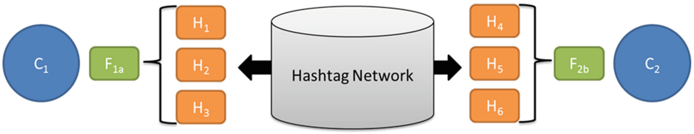


Fig. 4. Hashtag network and feature matching

3.7 Assign the Polarity of Opinions and Find Competition Behavior

In this section, we measure the opinion polarity for each extracted feature from component. In order to assign the polarity of opinion, we use the variation of point-wise mutual information and the Turney's [10] proposed method.

The variation of point-wise mutual information (V-PMI) apply to measure the strength between features and terms which have opinion polarity meaning.

$$PMI'\left(c_i, S, f_{i_j}\right) = frequence\left(S, f_{i_j}\right) * log_2 \frac{P\left(S, f_{i_j}\right)}{P(S) * P\left(f_{i_j}\right)}$$

- c_i: a specific component
- S: a set of opinion words (Adjective), $S = \{s_1, s_2, s_3 \ldots\}$
- f_{i_j}: a feature j of the component c_i

$$P\left(S, f_{i_j}\right) = \frac{frequence\left(Adjective\ s\ \in S\ \&\ f_{i_j}\ appear\ in\ one\ tweet\right)}{frequence\left(Adjective\ \&\ f_{i_j}\ appear\ in\ one\ tweet\right)}$$

And the polarity of the f_{i_j}:

$$Polarity\left(f_{i_j}\right) = PMI'\left(c_i, S_p, f_{i_j}\right) - PMI'\left(c_i, S_n, f_{i_j}\right)$$

- S_p: a set of positive opinion words (Adjective)
- S_n: a set of negative opinion words (Adjective)

4 Experiment

In this section, we show the performance for four specific queries. The answer set is annotated by human and is shown in Table 3.

Table 3. Dataset for evaluation

Query	Answer set of candidates
Google	Twitter, Apple, FB, Facebook, Microsoft, HP, Bing
Apple	Google, Microsoft, Motorola, Blackberry, Orange, HP
Android	iPhone, BlackBerry, iOS, BB, Symbian, iPad, MeeGo
Packer	Steelers, Bears

4.1 Candidate Extraction

Here, we extract the component candidate like we mentioned above section. Table 4 shows that use the PMI value to cut the candidate set (PMI cut) were better than only select the component by the appear frequency (Top 30).

Table 4. Candidate Extraction Threshold

Query		Precision	Recall
Google	Top 30	0.1666	0.625
	PMI cut	0.2	0.75
Apple	Top 30	0.1333	0.5
	PMI cut	0.2	0.75
Android	Top 30	0.2666	0.6666
	PMI cut	0.2333	0.5833
Packer	Top 30	0.1666	0.3571
	PMI cut	0.2	0.4286

4.2 Feature Extraction

Figure 5 shows that at the point P@10 almost every query can find its feature. The query "Packer" was different from others because people always use the word "GB Packer", "Green Bay Packer", "Superbowl Packer", "NFL Packer" and "Super Bowl Packer" to talk about "Packer".

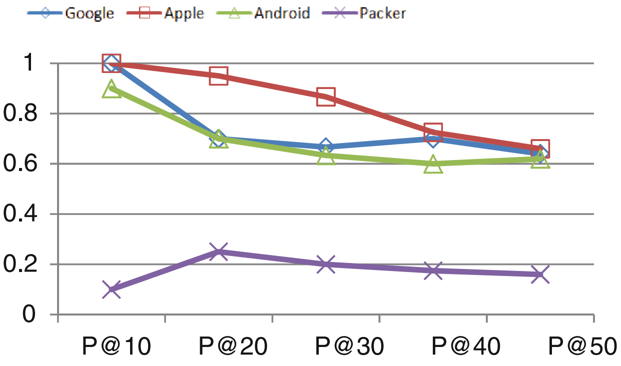


Fig. 5. Performance of Top-n precision

Our feature extraction approach will extract the noun or noun phrase which describes the query, these noun or noun phrases cannot be features. At the point P@50, almost every query's precision still remain 0.6 to 0.7.

4.3 Competition Component Extraction

We select 30 competition component candidates with PMI cut and evaluate its performance by using mean average precision. The high MAP value indicate that we can promote the rank of correct candidate.

In Table 5, each column means that:

Average CBS Score: Ranking the candidates by the average CBS score between the candidates.

Average Shortest Path: Ranking the candidates by the all features' average shortest path on the hashtag network between the candidates.

Aggregate: Ranking the candidates by normalize above two values and average it.

Table 5. Statistics in hashtag graphs

Query	Average CBS Score	Average Shortest Path	Aggregate
Google	0.2453	0.4830	0.3159
Apple	0.1520	0.2663	0.1682
Android	0.2239	0.3782	0.4385
Packer	0.3319	0.4513	0.4142

Average shortest path is an important feature to identify the correct candidate. Average CBS score should be another dominate feature, but our assign feature polarity approach will affect by the adjective opinion words which is not describe the feature. We can use the aggregate way to improve our final result.

5 Conclusion

In this paper, we introduce the competition behavior and competition component. It is different from traditional opinion mining way. Traditional opinion mining can only tell us how good or how bad about something. We mining the inside of the opinion, and find the feature and the competition component candidates. By using the hashtag network, we can solve the feature matching problem. Our initial indicate that the easy approach is useful to extract the candidate and the feature. Competition behavior can be identified through those approaches.

In future work, Twitter's other features (reply, retweet, mention, follow) can be used to find another way to extract the candidate. And we can use the distance between two words to improve the CBS score.

References

1. Wikipedia.: Microblogging, 18 December 2011 15:39 UTC. http://en.wikipedia.org/w/index.php?title=Microblogging&oldid=466348437
2. Wikipedia.: Twitter, 18 December 2011 15:44 UTC. http://en.wikipedia.org/w/index.php?title=Twitter&oldid=466256657
3. Liu, B., Hu, M., Cheng, J.: Opinion observer: analyzing and comparing opinions on the Web. In: Proceedings of the 14th International Conference on World Wide Web (2005)
4. O'Connor, B., Balasubramanyan, R., Routledge, B.R., Smith, N.A.: From tweets to polls: linking text sentiment to public opinion time series. In: Proceedings of the International AAAI Conference on Weblogs and Social Media (2010)

5. Chamlertwat, W., Bhattarakosol, P., Rungkasiri, T., Haruechaiyasak, C.: Discovering consumer insight from twitter via sentiment analysis. J. UCS **18**, 973–992 (2012)
6. Wikipedia.: Competition, 5 July 2012 17:10 UTC. http://en.wikipedia.org/w/index.php?title=Competition&oldid=499458477
7. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2004)
8. Hu, M., Liu, B.: Mining opinion features in customer reviews. In: Proceedings of the 19th National Conference on Artificial intelligence (2004)
9. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**, 39–41 (1995)
10. Turney, P.D.: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (2002)
11. Pang, B., Lee, L. Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, vol. 10 (2002)
12. Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., Smith, N.A.: Part-of-speech tagging for Twitter: annotation, features, and experiments. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers, vol. 2 (2011)

Social Network Community Detection Using Strongly Connected Components

Wookey Lee[1], James J. Lee[1], and Jinho Kim[2](✉)

[1] Informatics Laboratory, Department of I.E, INHA University,
253 Yong-Hyun Dong, Incheon 402-751, South Korea
{trinity,bigjameslee}@inha.ac.kr

[2] Department of Computer Science, Kangwon National University,
192-1 Hyoja-Dong, Chuncheon, Kangwon 200-701, South Korea
jhkim@kangwon.ac.kr

Abstract. The hasty information growth of social network poses the information searching efficiency trials for network mining research. Social network graphs and web graphs are huge sources of highly densely connected hypertext links so that the social networks can be described by a directed graph. This kind of network has inherent structural characteristics such as overly expanded, duplicated, connectedness, and circuit paths, which could generate serious challenges for structured searching for sub-network isomorphism and community detection. In this paper, an efficient searching algorithm is suggested to discover social network communities for overcoming the circuit path issue embedded in the social network environment. Experimental results indicate that the proposed algorithm has better performance than the traditional circuit searching algorithms in terms of the time complexity as well as performance criteria.

Keywords: Strongly connected component · Circuit path · Information communities

1 Introduction

Recently more and more information sources are represented as a network where the information will be richer than what the node only has been represented. For example, the Web pages have been processed by keyword analyses or matrix representation for each node; however the nodes are connected each other with hypertext links that can give much deeper level of information with link based algorithms like PageRank. Another example is Patent information where it also can be analyzed swallow levels without citation analysis. Much more information sources are engaged in graph based features such as news or blog, Wikipedia, Twitter, YouTube, Facebook, Transportation, Chemical components, Medical information, Bio-informatics, Supply Chain Management, Social Annotations, XML and RDF/RDFS/OWL with Semantic Web, AppStore and App Marketplace, etc. [2, 3, 6, 8, 10, 15, 18].

The Web also is a typical huge source of information and continues to grow explosively at a million pages per day. This leads to a large number of web pages being

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 596–604, 2014.
DOI: 10.1007/978-3-319-13186-3_53

retrieved for most queries, hence searching information on the Web is becoming an increasing difficult task. Accordingly, the importance for structuring the Web has been increasing. There are significant amount of hyperlinks in the Web, which comprise of the searching paths. Those paths, called ‘circuit’, can be traversed circularly. The circuits are inevitably existent in Web structure and can be considered as serious problems diminishing the effectiveness in information searching. In other words, the circuit is a web community within which the web pages are frequently accessible among them. In this paper, the efficient searching algorithm is suggested to find all the circuits embedded in the Web. It, however, is very hard to detect circuits since there are fundamentally huge numbers of web circuits in a web site, and there is a performance issue to isolate the duplicated web circuits [4, 9, 12, 15].

From a web-as-a-graph perspective, as shown in Fig. 1, a Social networkis suited to represent the Web, where a web page is set as a vertex and hyperlink as an edge that is widely accepted as the basic approach to analyze the Web structure [1, 5, 6, 8]. The Web structure can be easily constructed the search and storing data structure by using graphs.

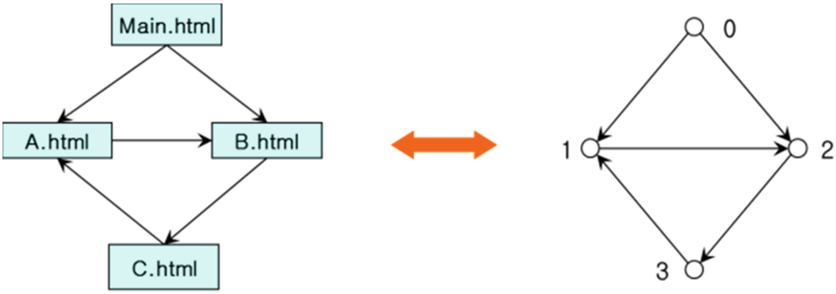


Fig. 1. The web can be described as a graph

According to our physical research on the web, we found that there are some interesting features in the web graphs such as a strongly connected component (*SCC*) and a tree structure. In Fig. 2, the node 1 can be a root node so called a homepage having a tree structure (i.e., nodes 2, 6, 7, 8), and two *SCC*s (i.e., a series of nodes 5, 12, 11 and 3, 4, 10, 9, respectively) etc. Note that the web communities can only possible within the circuits. However, on detecting the web circuits, conventional algorithms are very weak to differentiate the duplicate circuits. Note that a circuit with 5, 11, and 12 is the same circuit by 11, 12, and 5 and/or by 12, 5, and 11. In this paper, by detecting the *SCC*, an efficient searching for the web communities can be possible to find for overcoming the duplicate circuits embedded in the Web.

This paper is organized in the following way. Section 1 reviews the related basic definitions. Sections 2 and 3 introduce some definitions and the proposed searching algorithms. Section 4 presents and analyses our experiment. Finally, Sect. 5 concludes the paper.

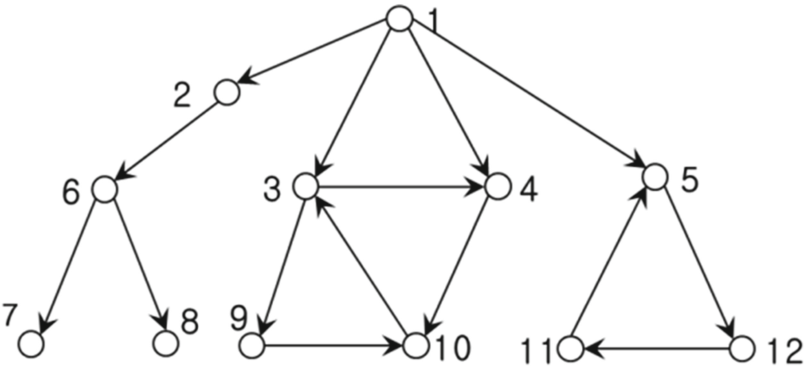


Fig. 2. The example web structure consists of web nodes and arcs

2 Social Network and Web Circuits with *SCC*

Social Network. Graph $G(V, E)$ consists of vertex V and edge E that connects between vertexes <v, w> for v, $w \in V$. Let P be the arbitrary web pages to be structured. The first page for starting searching is represented as $P_0 \in P$ and the linked sequence of pages is defined as $\{P_i\}$ $(i \in Z, i > \; = 0)$, where i is the page index. If $|P| = n(n \in \mathrm{Z}, n \neq 0)$, then $P = \{P_0, \ldots P_{n-1}\}$. The set of Web pages, P, can be described by Web graph, $G_p = (I_p, E_p)$, where I_p is the vertex of G_p, that is, the set of the index i. I_p is the set of vertices of G_p, hence can be defined as $I(G_p) = \{0, \ldots, n-1\}$. E_p is the edges of G_p and represents the links connecting two pages. Given two indices $i,j(i,j \in I_p)$, the link $<i,j> \; \in E_p$ means that an index i enters into the j. Since E_p represents the set of all links in $G_p, E(G_p) = \{<i,j>\}$, if $\forall <i,j> \; \in E_p$.

Circuit. When the Social network graph $G_p = (I_p, E_p)$ is searched, the *circuit* is defined as a sequence of the Web page indices $(i_{1,}, \ldots, i_k, k \le n)$ satisfying the following conditions: If $i_x, i_y \in I_p(1 \le x, y \le k, k \le 2)$ are indices, $E'_p = \{<i_1, i_x>, <i_x, i\ y>, <i_y, i_k>\}$ $(E'_p \subseteq E_p,)E'_p$ and $i_1 = i_k$, then $C_1 = i_1, i_x, i_y, i_k(C_1 \subseteq I(G_p), C_1 \subseteq C(G_p))$ is called a *circuit*, where $C(G_p)$ is the set of all circuits in G_p. In this case, $i_1 \to i_x \to i_y \to i_k$ is called the *circuit path.*

Repetition Circuit. The Social network graph $G_p = (I_p, E_p)$ can have the *repetition circuits* which are the circuits discovered repeatedly in the Web searching. Let be $|C| = k$ the length of the circuit path C. If $C_1 = i_1, i_2 \ldots, i_{k-1}, i_k(C_1 \subseteq C(G_p), |C_1| = k), C_2 = i_2, i_3 \ldots, i_k, i_1(C_2 \subseteq C(G_p))$, and $|C_2| = \mathrm{k}$, then C' and C are identical circuits. For example, one circuit 1 → 2 → 3 → 1 and another circuit 2 → 3 → 1 → 2 have the same path lengths and the order of connection patterns. If these circuits are discovered when searching, these two circuits are considered as repetition circuits (see Fig. 3).

In the social network environment, the more the links exist and the longer the paths will be, and the more frequent many repetition circuits may be derived. The repetition circuits should be avoided for which this paper will present an algorithm to remove such repetition circuits.

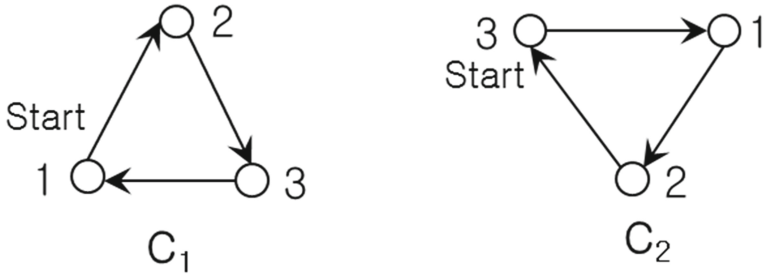


Fig. 3. The repetition circuits

SCC. A Strongly Connected Component is a maximal subgraph that exist circuit paths among all nodes in the subgraph. The graph G is an *SCC*, if there is a path between v and w $(p_1 : v \rightarrow w, p_2 : w \rightarrow v)$ [13, 16]. For example, there are two *SCC*s in Fig. 3, such as $3 \rightarrow 4 \rightarrow 9 \rightarrow 10$, and $5 \rightarrow 12 \rightarrow 11$. In the first *SCC*, there are multiple circuits in an *SCC*.

Theorem 1. In the *SCC* of graph G, there exists a circuit.

Proof: Assume that a graph G does not have an *SCC*. Unless the graph has an *SCC*, there is no path from a node in G that returns to the node again. Since there is no path from a node to the same node, so no circuit exists. By contradiction, the theorem is proved.

Theorem 2. The number of the circuits in *SCC* is the same as the graph G.

Proof: Assume that a graph G has two *SCC*s, called *SCC*1 $(|SCC_1| = m)$ and *SCC*2 $(|SCC_2| = n)$, and there is an *SCC* that includes the two *SCC*s. If there is a path that connects the two *SCC*s, then the two *SCC*s are included in a big *SCC*, which is a contradiction that there are two *SCC*s. There is no circuit outside each *SCC*, therefore the number of circuits in graph G is the same as the sum of the number of circuits of each *SCC*. So that $|SCC_1| + |SCC_2| = m + n$.

3 Circuit Searching Algorithms

The basic algorithm for circuit searching is backtracking approach based on depth-first search [5, 13, 17]. In depth-first search, edges are explored out of the most recently discovered vertex *v* that still has unexplored edges leaving it. When all the edges of *v*'s have been explored, the search backtracks to explore edges leaving the vertex from

```
procedure CircuitDetectionWith$SCC$_init()
begin
  // visited[] := false,  possibility[] := false, stack[] := false
  for (i = 1 to v)
    begin
      if (roots[i] = true)
      begin
        root := AdjacencyList[i];
        CircuitDetectionUsing$SCC$(root);
      end
    end
end

procedure CircuitDetectionUsing$SCC$(Node v)
begin
    visited[v.key] := true;
    Push(v.key);
    for (each vertex w adjacent from v)
    begin
      if (w.key ≠ root.key)
      begin
        if (visited[w.key] = true)
        begin
          if (possibility[w.key] = false)
          begin
            continue;
          end
          //Enumerating Circuit Path;
          continue;
        end
        CircuitDetectionUsing$SCC$(w);
      end
      else
      begin
        //Enumerating Circuit Path;
        continue;
      end
    end
    visited[v.key] := false;
    possibility[pop()] := false;
end
```

Fig. 4. Circuit Detection Algorithm (CduSCC)

which *v* was discovered. This process continues until we have discovered all the vertices that are reachable from the original source vertex. If any undiscovered vertices remain, then one of them is selected as a new source and the search is repeated from that source. This entire process is repeated until all vertices are discovered.

The graph can be represented using the adjacent list, where each vertex is defined as node. Each node is composed of value and the pointer pointing to the next node. The value here indicates the index of Web page and the pointer the edges incident to the next vertex [11, 18]. The first mode in adjacent list is the head node which can be the list of all the node indices.

Figure 4 shows the pseudo-code of the basic circuit searching algorithm based on *SCC* search. If the number of vertices is V and the number of edges is E, then the normal running time for searching the vertices is $O(V)$ and visiting the edges $O(E)$ and the total cost is $O(V + E)$ in a diagraphs without circuits. Whenever the circuits are found, the time complexity is increased to $O(VEC)$, where C is the number of circuits. Hence the total cost is $O(EV(C + 1))$. Since it compares all vertices in stack with the newly pushed vertices, the running time is $O(EV(C - 1)/2)$. It has a drawback for searching the multiple repetition circuits.

All vertices are examined to confirm in the conventional approaches that have to visit even after a circuit searching, that is executed again from the unvisited vertices [10, 13]. Whenever the circuits are found, the running time needs to be taken because of the comparison with all the vertices in the stack.

The searching algorithm proposed in this paper finds only the circuits discovered from the starting vertices based on depth-first search. Then the pre-visited vertices are ignored for further searching to eliminate the repetition circuits. The starting vertex is to be the first vertex of a circuit, which results in the time complexity $O(1)$.

4 Experiment

The proposed algorithms are examined using Tiernan and Weinblatt model [14]. The RemoveN method detects all the theoretically possible circuits but repetition circuits in fully connected graphs. The conventional method, named Backtrack, on the other hands, is confirmed that the repetition circuits are getting more detected when the

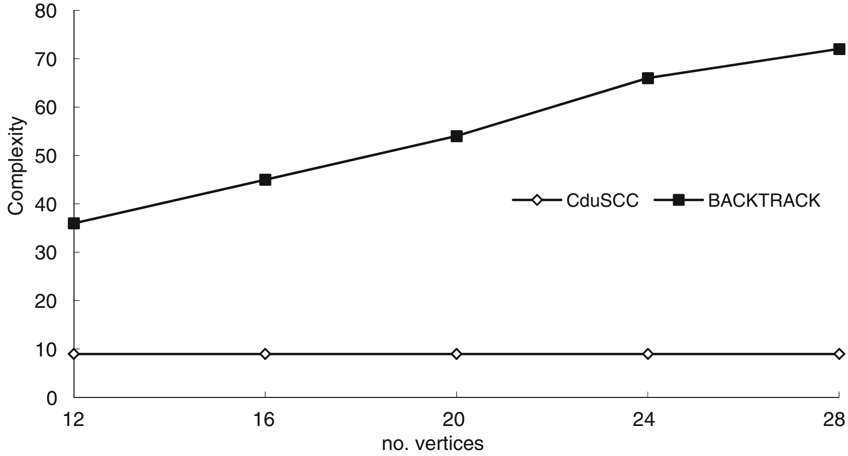


Fig. 5. Complexity analysis with respect to the number of vertex

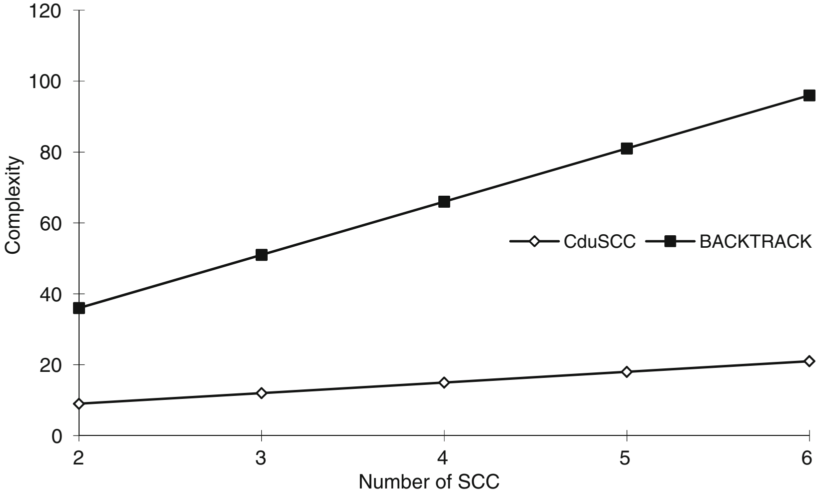


Fig. 6. Complexity analysis with respect to the number of *SCC*

number of vertices is increasing and the comparative performance is getting worse. As the number of vertices is increasing, the number of detected circuits is shown to increase rapidly.

The algorithm may generate exponentially more circuits by increasing number of edges leaving from a particular vertex. Moreover, it will face with more repetition circuits. Therefore it will take more execution time because the detected circuits should be compared with each other to eliminate the repetition circuits. In this respect, the proposed algorithm will be very effective in the social network structure, where the vertices and edges are growing explosively. We also apply the proposed algorithm to the google-watch.org site using the AnchorWoman system [7], which has been developed using C# in Compaq server in order to test the performance. Figure 7 shows the target nodes and links found and the detailed nodes are summarized. In Figs. 5 and 6, the complexity analyses have be done in terms of the increment of the number of vertices and those of *SCC*, where our algorithm outperformed conventional backtrack approaches which means that the more the number of *SCC*, the better the performance expectation. We can conclude that our approach can generate much advanced results without sacrificing system consumption such as CPU and memory.

We experimented on the real world environment with the hurricane data center in Louisiana State University (http://hurricane.lsu.edu) where about 4.3 million nodes and 16 trillion edges are derived. The *SCC*s are generated which can be called as an information community inside of the graph, and interestingly some form of Trees are generated which may be an information guideline even if it is not yet thoroughly analyzed, since it is out of our focus to find out the *SCC*s. The sample results are represented in Fig. 7 and Table 1 respectively.

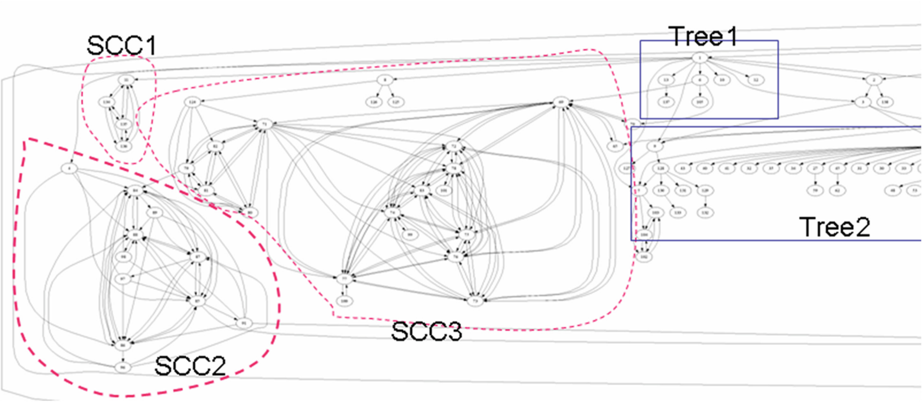


Fig. 7. Nodes and arcs with *SCC*s and Tree structure in a part of hurricane web site

Table 1. Node and the corresponding URL for hurricane web site

Node id	Path URL
Node0	http://hurricane.lsu.edu
Node1	http://hurricane.lsu.edu/navbar.htm
Node2	http://hurricane.lsu.edu/home_page.htm
Node3	http://hurricane.lsu.edu/newsbriefs.htm
Node4	http://hurricane.lsu.edu/personnel.htm
Node5	http://hurricane.lsu.edu/research.htm
Node6	http://hurricane.lsu.edu/publications.htm
Node7	http://hurricane.lsu.edu/academic_programs.htm
……	……
Node138	http://hurricane.lsu.edu/mission_statement.htm
Node139	http://hurricane.lsu.edu/katrinacontact.htm

5 Conclusion

The network graph algorithm is applied to transform the network structure into directed graph, where a strongly connected component can be derived. We have introduced a new algorithm for structuring the social community through finding out and eliminating the repetition circuits for a graph environment. Traditional algorithms dealing with the circuit generation issues have been not enough efficient in terms of the time complexity, in this paper, however, our approach proved to be polynomial. Therefore the proposed algorithm presented as a very efficient with respect to the number of nodes and arcs as well as the number of *SCC*s. In the future research, our approach can be a strong alternative for the social network analysis or the Web search agent, where the real-time searching performance can be a high priority. Moreover it can be applied to the various graph environments such as bio-informatics, social activity networks, patent graphs, data graphs, vehicle trajectory graphs, etc.

Acknowledgement. This work was supported by Defense Acquisition Program Administration and Agency for Defense Development under the Contract UD080042AD, Korea.

References

1. Amer-Yahia, S., Huang, J., Yu C.: Building community-centric information exploration applications on social content sites. In: SIGMOD, pp. 947–952 (2009)
2. Arora, N.R., Lee, W.: Graph based ranked answers for keyword graph structure. New Gener. Comput. **31**(2), 115–134 (2013)
3. Chakrabarti, S., Chakrabarti, S., Pathak, A., Gupta, M.: Index design and query processing for graph conductance search. VLDB J. **20**(3), 445–470 (2011)
4. Hajibagheri, A., Alvari, H., Hamzeh, A., Hashemi, S.: Community detection in social networks using information diffusion. In: ASONAM, pp. 702–703 (2012)
5. Lee, W., Loh, W., Sohn, M.: Searching Steiner trees for social network query. CANDIE **62**(3), 732–739 (2012)
6. Lee, W., Leung, Carson K.-S., Lee, J.: Mobile web navigation in digital ecosystems using rooted directed trees. IEEE TIE **58**(6), 2154–2162 (2011)
7. Lee, W., Lee, J., Kim, Y., Leung, C.: AnchorWoman: top-k structured mobile web search engine. In: CIKM, pp. 2089–2090 (2009)
8. Maserrat, H., Pei, J.: Community preserving lossy compression of social networks. In: ICDM, pp. 509–518 (2012)
9. Nivasch, G.: Circuit detection using a stack. Inf. Process. Lett. **90**, 135–140 (2004)
10. Korula, N., Lattanzi, S.: An efficient reconciliation algorithm for social networks. PVLDB **7**(5), 377–388 (2014)
11. Rautenbach, D., Szwarcfiter, J.: Unit interval graphs of open and closed intervals. J. Graph Theor. **72**(4), 418–429 (2013)
12. Stevens, B., Williams, A.: Hamilton cycles in restricted and incomplete rotator graphs. J. Graph Algorithms Appl. **16**(4), 785–810 (2012)
13. Tarjan, R.: Depth-first search and linear graph algorithms. In: FOCS, pp. 114–121 (1971)
14. Qi, X., Tang, W., Wu, Y., Guo, G., Fuller, E., Zhang, C.-Q.: Optimal local community detection in social networks based on density drop of subgraphs. Pattern Recogn. Lett. **36**, 46–53 (2014)
15. Xie, J., Szymanski, B.K.: Towards linear time overlapping community detection in social networks. In: PAKDD, pp. 25–36 (2012)
16. Zhang, X., Cheng, J., Yuan, T., Niu, B., Lu, H.: TopRec: domain-specific recommendation through community topic mining in social network. In: WWW, pp. 1501–1510 (2013)
17. Zhou, Y., Liu, L.: Social influence based clustering of heterogeneous information networks. In: KDD, pp. 338–346 (2013)
18. Zhu, Y., Zhong, E., Pan, S., Wang, X., Zhou, M., Yang, Q.: Predicting user activity level in social networks. In: CIKM, pp. 159–168 (2013)

Data Mining in Biomedical informatics and Healthcare

Automatic Segmentation and Quantitative Analysis of Gray Matter on MR Images of Patients with Epilepsy Based on Unsupervised Learning Methods

Rui Wang[1], Jie Wang[1], Jing Ding[2](✉), and Su Zhang[1](✉)

[1] School of Biomedical Engineering,
Shanghai Jiao Tong University, Shanghai, China
suzhang@sjtu.edu.cn

[2] Neurology of Department, Zhongshan Hospital,
Fudan University, Shanghai, China
dingjing1018@sina.com

Abstract. The quantitative analysis of volume information about gray matter (GM) on magnetic resonance (MR) images is important in both research and clinical diagnosis of patients with epilepsy. In this paper, a k-means method and an expectation maximization algorithm are implemented respectively to achieve segmentation of GM on MR images at the transverse and coronal plane. The experiments were performed on both multi-modal and mono-modal MR images and the similarity index values for the accuracy of automatic segmentation with manual segmentation were consistently high for patients with epilepsy (transverse plane: 0.806; coronal plane: 0.837). The results demonstrated that the automatic segmentation methods implemented in this paper are accurate and efficient to realize extraction of GM of patients with epilepsy in both transverse and coronal plane.

Keywords: Epilepsy · Automatic segmentation · K-means · Expectation maximization

1 Introduction

Epilepsy is a common neurological disorder, often accompanied by convulsions, foaming at the mouth and other symptoms [1]. The pathological mechanism of epilepsy is unclear but it is suggested to be associated with brain trauma, stroke, brain cancer, drug misuse, and alcohol abuse. Epilepsy affects about 1 % of people worldwide and the vast majority of patients need long-term use of medications to control seizures. The treatments of epilepsy depend on doctors' subjectivity and experience most of the time because the doctors determine drug dosages during treatment according to verbally description of the symptoms. Clinical studies indicate structural gray matter abnormalities exist on brain MR images of patients with epilepsy and volume changes of gray matter (GM) may provide the foundation of physiological status for GM [2]. Thus quantitative volumetric information helps doctors diagnose and cure the epilepsy better.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 607–615, 2014.
DOI: 10.1007/978-3-319-13186-3_54

Volume measurement of GM requires segmentation performed in advance thereby the structure of GM distinguished from other tissues clearly. Image segmentation is the process of partitioning a digital image into multiple non-overlapping regions according to the similarity or difference between regions. Segmentation of GM aims to extract the structure of GM from the whole brain region, reveal the etiology of epilepsy, and investigate the pathogenesis in further. The techniques employed in traditional image segmentation include threshold methods, grey clustering, edge detection and region extraction. In consideration of general characteristics of medical images such as artificial noise, low contrast, inhomogeneity, and blurred boundary, segmentation of brain varies from supervised methods to unsupervised methods. Supervised methods are able to obtain more accurate models whereas the training is lengthy, labor-intensive and sometimes the results tend to migrate from the standard ones. By contrasts, unsupervised methods employ certain models to fit different types of data without label and the models can be solved by numerical iteration method, etc. Unsupervised methods are widely used in automatic segmentation of medical images [3], including fuzzy c-means (FCM), expectation maximization (EM) algorithm, k-means, etc. K-means method is faster than FCM and EM when performing segmentation on multimodal medical images whereas it is not good at dealing with fuzzy border caused by partial volume effect. Gaussian mixture model (GMM) models the gray distribution of brain MR images well [4] and parameters in the model can be estimated by EM algorithm via maximum likelihood estimation (MLE).

In this paper, k-means clustering algorithm is utilized to perform GM segmentation using two modalities of MR images of patients with epilepsy, namely, T1 fluid attenuated inversion recovery (FLAIR) and double inverse recovery (DIR) images. In addition, GMM is employed to model the gray distribution of different brain tissues on T1 fast spoiled gradient echo (FSPGR) images. The performances of different automatic segmentation methods are evaluated in comparison with results of manual segmentation, which are consider as the ground truth in medical image segmentation.

2 Methods

2.1 Image Preprocessing

Image preprocessing is necessary before the segmentation of GM to minimize the effect of image artifacts and align different MR sequences in the same space. Three processing steps are generally considered in automatic segmentation of GM, which are intensity inhomogeneity correction, registration, and brain extraction.

- **Intensity inhomogeneity (IIH) correction:** The IIH usually refers to the slow, nonanatomic intensity variations of the same tissue across the medical images due to inhomogeneity of the static or radio-frequency nonuniformity, which can substantially reduce the accuracy of segmentation. Thus, IIH correction was realized by N3 inhomogeneity correction module [5] in MIPAV software (iterations = 100, end tolerance = 0.001) in this paper.
- **Registration:** Image registration [6] is demanded when combining information of multimodal MR sequences to achieve segmentation of GM. It determines a

transformation which makes different modalities of MR images into the same space. An optimized automatic registration method in MIPAV software was employed to perform registration between T1 FLAIR and DIR images. The optimized automatic image registration is a non-rigid registration method, and it determines a transformation by minimizing a cost function, which is evaluated at different image resolutions from the lowest to highest. Each step of increasing resolution is performed by setting the previously determined optimal transformation as initial value and optimizing the value continuously until the performance of registration achieve satisfactory effect.

- **Brain Extraction:** To avoid non-brain tissue including skull and scalp signals disturbing segmentation of GM, a FSL's Brain Extraction Tool (BET) [7] was employed to constrain the segmentation task performed only on the remaining brain voxels.

2.2 K-means Algorithm

K-means algorithm [8] is an iterative algorithm of clustering which is simple, straightforward, and adaptable. It can perform segmentation of multimodal images quickly and efficiently. Let Ω denotes a set of voxels in a d dimensional space, namely, $\Omega = \{x_i | i = 1, 2, \ldots, N\}$, and vector $x_i = \{x_i^1, \ldots, x_i^d\}$ is a sample in Ω. The aim of K-means algorithm is to construct k clusters so that all the samples should be assigned to different clusters. The centers of different clusters are represented as $c_j, j = 1, \ldots, k$. All the voxels of images are assigned to their own classes according to similarity. Euclidean distance method is often used to find the similarity and the procedure of clustering is performed by minimizing a cost function iteratively. The cost function can be expressed as:

$$Cost = \sum_{i=1}^{N} (\arg\min_j \left\| x_i - c_j \right\|_2{}^2) \tag{1}$$

The concrete realization process of K-means algorithm is illustrated as follows:

Step 1: Set of the number of clusters k and initialize starting value of clustering centers $c_j, j = 1, \ldots, k$.

Step 2: Reassign samples in Ω to closest cluster center and update cluster label (CL) such that CL_i is cluster ID of ith voxel in Ω.

Step 3: Update the clustering centers c_j, $j = 1, \ldots, k$ by computing means of samples in jth cluster.

Repeat step 2 and step 3 until the convergence of cost function given by Eq. (1).

2.3 EM Segmentation

Gaussian mixture model (GMM) [9] is the most commonly used model for statistical segmentation of brain MR images. It is a parametric probabilistic density model that assumes all the data points are generated from a mixture of finite number of Gaussian distribution components. The probability density function (PDF) of GMM is given by

$$\varphi(y_i; z_i) = \sum_{i=1}^{k} \pi_j \psi(y_i; x_i, \mu_j, \Sigma_j) \tag{2}$$

where k denotes the number of single Gaussian component, namely, the number of clusters. y_i denotes the gray intensity of an arbitrary voxel in MR image, and z_i is the corresponding estimated clustering label of y_i. ψ is the PDF of single Gaussian component in GMM with mean μ_j, variance Σ_j, and proportion of the jth class. The vector of all the unknown parameters is declared to be

$$\Theta = (z_i, \mu_i, \Sigma_i, \pi_i) \tag{3}$$

The unknown parameters can be estimated by MLE and ML parameters estimates can be obtained using an expectation-maximization (EM) algorithm. The log likelihood of the data set is given by

$$\log L(\Theta) = \sum_{i=1}^{n} \log\{\sum_{j=1}^{k} \pi_j \psi(y_i; x_i, \mu_j, \Sigma_j)\} \tag{4}$$

EM algorithm is implemented by a two-step iterative method, namely, the expectation step (E-step) and the maximization step (M-step). The E-step is performed on a Q function on the $(k+1)$th iteration of EM algorithm and computes

$$Q(\Theta|\Theta^{(k)}) = \mathrm{E}\left[\log L(\Theta)|y, \Theta^{(k)}\right] \tag{5}$$

where $\mathrm{E}\left[\log L(\Theta)|y, \Theta^{(k)}\right]$ is the expectation of the complete-data log likelihood. The M-step updates the estimate of $\Theta^{(k+1)}$ to maximize $Q(\Theta|\Theta^{(k)})$ as following

$$\Theta^{(k+1)} = \arg\max Q(\Theta|\Theta^{(k)}) \tag{6}$$

In consideration that the initialization of starting value of unknown parameters affects the performance of EM algorithm, k-means clustering algorithm is often used to initialize unknown parameters in EM algorithm.

3 Experiments and Results

The experiments were performed on images of 4 patients (two males and two females) diagnosed with epilepsy, who were participating in research studies at Shanghai Zhongshan Hospital. This study was approved by the Institutional Review, and written informed consent was obtained from all patients. All images were acquired on the same

3T MR scanner (GE MEDICAL SYSTEMS), including a 3D T1 FLAIR sequence (TR = 1786.6 ms, TE = 26.328 ms, TI = 860 ms), a 3D DIR sequence (TR = 15002 ms, TE = 85.176 ms, TI = 2950 ms), and a 3D T1 FSPGR sequence (TR = 6.932 ms, TE = 2.248 ms, TI = 700 ms). The automatic segmentation of GM in patients with epilepsy was classified into two independent processes according to different modality of MR images employed by automatic segmentation.

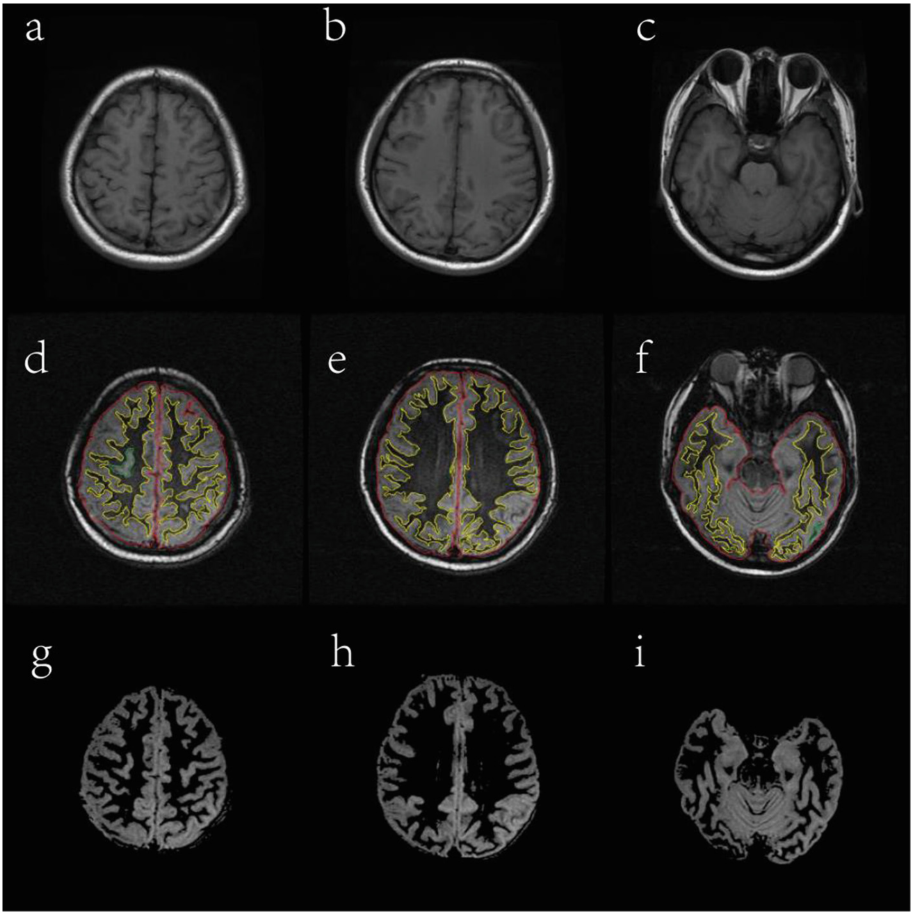


Fig. 1. Segmentation results of GM on three slices at the transverse plane using k-means method. (a)–(c) original T1 FLAIR images. (d)–(f) results obtained by manual segmentation on DIR images; the red, yellow and green line indicate outer contour, inner contour, and closed contour of GM respectively. (g)–(i) results of automatic segmentation (Color figure online).

3.1 K-means Segmentation on T1-FLAIR and DIR

T1 FLAIR is a kind of rapid inversion recovery spin echo sequence with fast imaging speed and good contrasts between GM and white matter (WM). DIR sequence combines two inversion pulses to suppress signals of WM and cerebrospinal fluid (CSF), making GM clearly stands out. The combination of T1 FLAIR sequence and DIR sequence provides more information when perform segmentation of GM and makes the border detected of GM more robust to noise.

K-means algorithm is capable of attaining fast segmentation of different regions of interest (ROI) using multi-modality images. As mentioned before, the MR images of the two modes, namely, T1 FLAIR and DIR, were corrected by intensity inhomogeneity correction module. Then the reference image set and floating image set were constructed by choosing images from DIR and T1 FLAIR, respectively. Non-rigid registration was performed to match the floating images to the reference images and BET tool mentioned before was employed to remove non-brain tissue. After the image preprocessing, k-means segmentation was implemented on the two-modality images and the results were shown in Fig. 1. It can be found that the results are well in accordance with results of manual segmentation performed by a neurologist and radiologist referring to the corresponding T1 FLAIR and DIR images.

The volumes of GM of different patients can be figured out based on results of k-means segmentation and the information of GM volume will be useful for diagnosis of epilepsy by doctors. It is an observable fact that localizations of k-means segmentation exist as follows:

1. K-means segmentation may be sensitive to noises in DIR images, making the extracted GM affected by noise to some degree.
2. K-means algorithm sometimes has a drawback of local minimum problem, and segmentation more than once may be needed to settle this issue.

3.2 EM Segmentation on T1 FSPGR

T1 FSPGR sequence is able to implement fast imaging of brain and the contrast of GM in T1 FSPGR in comparison with WM is apparent. Operations of intensity inhomogeneity correction and brain extraction were performed before the EM segmentation. The results of EM segmentation and manual segmentation are shown in Fig. 2. It is not hard to find that the GM is well segmented by EM algorithm. The volume of GM in brain can be computed to provide doctors more information for their diagnosis with epilepsy. In fact, a deep segmentation of hippocampus in the coronal plane is of great sense for research of epilepsy.

3.3 Performance Evaluation of Automatic Segmentation

The results of automatic segmentation are compared with those of manual segmentation, namely, the gold standard. The similarity index (SI) and extra fraction (EF) [10] are calculated for selected slices, respectively. The SI are used to comparing the similarity between results obtained by automatic and manual segmentation while the

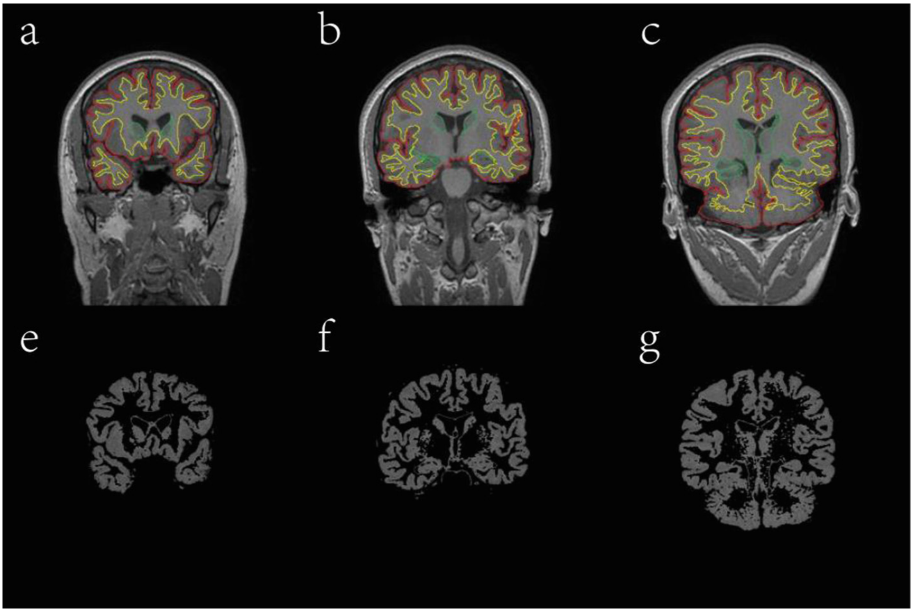


Fig. 2. Segmentation results of GM on three slices at the coronal plane using EM algorithm. (a)–(c) results of manual segmentation performed on original T1 FSPGR images; the red, yellow and green line indicate outer contour, inner contour, and closed contour of GM respectively. (d)–(f) results obtained by manual segmentation on DIR images. (g)–(i) results of automatic segmentation using k-means method (Color figure online).

EF are used to measure the extent of false positive (FP) existing in the results of automatic segmentation. The SI and EF are computed as follows:

$$\mathrm{SI} = \frac{2 \times (\mathrm{M} \cap \mathrm{A})}{\mathrm{M} + \mathrm{A}} \tag{7}$$

$$\mathrm{EF} = \frac{!\mathrm{M} \cap \mathrm{A}}{\mathrm{M}} \tag{8}$$

where M denotes the volume of GM segmented using manual method and A is the volume of GM obtained by automatic method. M ∩ A represents the volume of correctly classified voxels while !M ∩ A corresponds to areas misclassified by automatic method, namely, the false positives. The SI indicates a good segmentation when its value is close to 1. Average value of SI and EF were computed using results of automatic and manual segmentation on four slices of each patient and the results of different patients are presented in Table 1. The results demonstrate high similarities and low FP rates between results of automatic and manual segmentation.

Table 1. Similarity measurement comparison between results of automatic segmentation and manual segmentation

Patients	SI_kmeans	EF_kmeans	SI_EM	EF_EM
Patient I	0.845	0.171	0.852	0.196
Patient II	0.765	0.224	0.825	0.221
Patient III	0.786	0.256	0.814	0.235
Patient IV	0.828	0.198	0.856	0.166
Average	0.806	0.212	0.837	0.205

4 Discussion

In this paper, GM of patients with epilepsy was segmented on images at the transverse and coronal plane, respectively. K-means algorithm was used to perform segmentation of GM at the transverse plane by combining information of T1 FLAIR and DIR sequences. EM segmentation was implemented on T1 FSPGR images at the coronal plane to extract GM in brain. Both of the two methods achieve good segmentation of GM. In fact, k-means algorithm is computationally faster than EM algorithm if only a classification is need. GMM based EM segmentation is good at modeling gray distribution of different brain tissues including WM, GM, and CSF. In further, fuzzy information is employed in EM segmentation while it is not considered in k-means method. The results of automatic segmentation were compared with those of manual segmentation and the performances were demonstrated well. The volumes of GM of patients with epilepsy can be figured out to provide doctors with diagnostic basic for clinical treatment. Our ongoing and further work includes automatic segmentation and quantification measurement of hippocampus in MR images of patients with epilepsy.

Acknowledgement. This research is supported by, National Basic Research Program of China (973 Program, No. 2010CB732506), NSFC (No. 81301213), NSFC (No. 81000609), NSFC (No. 60972110), and Major Scientific Project of Social Science Foundation of China (No. 11&ZD174). The authors would like to thank Dr. He Wang in Philips Healthcare, China for his help on MR.

References

1. Banerjee, P.N., Filippi, D., Allen Hauser, W.: The descriptive epidemiology of epilepsy—a review. Epilepsy Res. **85**(1), 31–45 (2009)
2. Bonilha, L., Edwards, J.C., Kinsman, S.L., et al.: Extrahippocampal gray matter loss and hippocampal deafferentation in patients with temporal lobe epilepsy. Epilepsia **51**(4), 519–528 (2010)
3. Balafar, M.A., Ramli, A.R., Saripan, M.I., Mashohor, S.: Review of brain MRI image segmentation methods. Artif. Intell. Rev. **33**(3), 261–274 (2010)
4. Zhang, Y., Brady, M., Smith, S.: Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. IEEE Trans. Biomed. Imaging **20**(1), 45–57 (2001)

5. Sled, J.G., Zijdenbos, A.P., Evans, A.C.: A nonparametric method for automatic correction of intensity nonuniformity in MRI data. IEEE Trans. Med. Imaging **17**(1), 87–97 (1998)
6. Zitova, B., Flusser, J.: Image registration methods: a survey. Image Vis. Comput. **21**(11), 977–1000 (2003)
7. Smith, S.M.: Fast robust automated brain extraction. Hum. Brain Mapp. **17**(3), 143–155 (2002)
8. Moftah, H.M., Azar, A.T., Al-Shammari, E.T., Ghali, N.I., Hassanien, A.E., Shoman, M.: Adaptive k-means clustering algorithm for MR breast image segmentation. Neural Comput. Appl. **24**, 1917–1928 (2014)
9. Shiee, N., Bazin, P.-L., Cuzzocreo, J.L., Blitz, A., Pham, D.L.: Segmentation of brain images using adaptive atlases with application to ventriculomegaly. In: Székely, G., Hahn, H.K. (eds.) IPMI 2011. LNCS, vol. 6801, pp. 1–12. Springer, Heidelberg (2011)
10. Anbeek, P., Vincken, K.L., van Osch, M.J., Bisschops, R.H., van der Grond, J.: Probabilistic segmentation of white matter lesions in MR imaging. NeuroImage **21**(3), 1037–1044 (2004)

An Author Topic Analysis of Tobacco Regulation Investigators

Ding Cheng Li[1](✉), Janet Okamoto[2], Scott Leischow[2], and Hongfang Liu[1]

[1] Mayo Clinic, Rochester, MN, USA
dingcheng@mayo.edu
[2] Mayo Clinic, Phoenix, AZ, USA

Abstract. To facilitate the implementation of the Family Smoking Prevention and Tobacco Control Act of 2009, the Federal Drug Agency (FDA) Center for Tobacco Products (CTP) has identified research priorities under the umbrella of tobacco regulatory science (TRS). As a newly introduced field, the current landscape of TRS research is unclear. In this work, we conducted a bibliometric study of TRS research by applying author topic modeling on MEDLINE citations published by currently-funded TRS principle investigators. Our initial results show that author topic modeling can address the issue of research interests reasonably. Furthermore, a network involving authors, topics and words can be established for more detailed bibliometric analysis. This network may also be useful to grantees and funding administrators in suggesting potential collaborators or identifying those that share common research interests for data harmonization or other purposes.

1 Introduction

To facilitate the implementation of the Family Smoking Prevention and Tobacco Control Act of 2009, the Federal Drug Agency (FDA) Center for Tobacco Products (CTP) was formed to oversee tobacco regulatory activities. Its responsibilities include setting performance standards, reviewing premarket applications for new and modified risk tobacco products, requiring new warning labels, and establishing and enforcing advertising and promotion restrictions. In order to meet these responsibilities, the CTP has identified research priories for tobacco regulatory science (TRS) in order to inform and guide the CTP's regulatory decision-making. While tobacco researchers have been examining some of the CTP's TRS research priorities for many years, they have not necessarily been doing so under the umbrella or specific title of 'tobacco regulatory science'. Therefore, examining and identifying research topics from the corpus of TRS work could help to more clearly define this growing research area. In this paper, we applied author topic modeling [1], a variation of Latent Dirichlet Allocation (LDA), to simultaneously model the content of documents and the interests of authors. Namely, given the broader TRS research field, we attempted to discover topics as well as general research interests utilizing MEDLINE citations for currently funded TRS investigators.

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 616–627, 2014.
DOI: 10.1007/978-3-319-13186-3_55

In the following, we introduce the background information about author topic modeling and then describe our experimental methodology as well as results and interpretation of our analyses.

2 Background

LDA is known for its ability to model document contents as a mixture of topics (which comprise words describing similar things). This results in improvements in the study of hidden semantics of documents compared with previous models like Latent Semantic Indexing (LSI) [2], probabilistic LSI [3], vector semantics [4] and so on. Modeling interests of authors is in fact not new in the bibliometric research. As early as 1999, McCallum proposed a mixture author model with the mixture weights for different topics fixed [5]. Then, in 2004, Rozen-Zvi proposed author topic modeling [1], which is the integration of LDA and the author model. It aims at extracting information about authors and topics from large text collections simultaneously. Since then, author topic modeling has been widely used in applications such as bibliometrics analysis [6], information extraction [7], social network analysis [8] named entity recognition [9] and MeSH indexing interpretation [10].

However, modeling author-topic-words relations in tobacco regulatory science has not been done so far. Therefore, our work aims at filling this gap so as to extend author topic models into medical corpus analysis. We believe that this research will be beneficial for advanced information retrieval and will inform the emerging field of tobacco regulatory science.

3 Materials and Methods

3.1 Author Topic Modeling

Author topic modeling is a variation of LDA, aiming to extract information about authors and topics from a large text collection simultaneously. It is a class of Bayesian graphical model for text document collections represented by bag-of-words. In the standard LDA, each document in the collection of *D* documents is modeled as a multinomial distribution over *T* topics, where each topic is a multinomial distribution over *W* words and both sets of multinomial are sampled from a Dirichlet distribution.

Different from LDA, author topic modeling incorporates authors by adding one more variable, which is uniformed assigned by a set of authors, an observed set in some corpus. As in LDA, a topic is chosen from a distribution over topics specific to that author, and the word is generated from the chosen topic.

To learn the model parameters, we use Gibbs sampling where the equation for author topic modeling is,

$$
\begin{aligned}
&P\big(z_{id} = t, y_{id} = a | x_{id} = w, \boldsymbol{z}^{\neg id}, \boldsymbol{y}^{\neg id}, \mathrm{A}, \alpha, \beta\big) \\
&\qquad \propto \frac{N^{WT}_{wt,\neg id} + \beta}{\sum_{w'} N^{WT}_{w't,\neg id} + W\beta} \frac{N^{TA}_{ta,} + \alpha}{\sum_{t'} N^{TA}_{t'\alpha,\neg id} + T\alpha}
\end{aligned}
$$

where, α and β are Dirichlet priors for topic distributions, $z_{id} = t$ and $y_{id} = a$ are the assignments of the ith word in document d to topic t and author a respectively and $x_{id} = w$ indicates that the current observed word is word w. N^{TA} represents the topic-author count matrix, where $N^{TA}_{ta,\neg id}$ is the number of words assigned to topic t for author a excluding the topic assignment to word w_{id}. Similarly, N^{WT} is the word-topic count matrix, where $N^{WT}_{wt,\neg id}$ is the number of words from w^{th} entry in the vocabulary assigned to topic t excluding the topic assignment to word w_{id}. Finally, $\mathbf{z}^{\neg id}$ and $\mathbf{y}^{\neg id}$ represent the vector of topic assignments and vector of author assignment in all corpus except for the i^{th} word of the d^{th} document respectively.

Following the same convention, the posterior distribution of θ_{ta}, the topic distribution of each document and ϕ_{wt}, the topic distribution of each word, can be estimated with the following equations where D refers to the corpus.

$$\begin{aligned}
\theta_{ta} = p(t|a,d) &= E[\theta_{ta}|\mathbf{z}^{\neg id}, D, \alpha] \\
&= \frac{N^{TA}_{ta,\neg id} + \alpha}{\sum_{t'} N^{TA}_{t'\alpha,} + T\alpha} \\
\phi_{wt} = p(w|t) &= E[\phi_{ta}|\mathbf{z}^{\neg id}, D, \beta] \\
&= \frac{N^{WT}_{wt,} + \beta}{\sum_{w'} N^{WT}_{w't,} + W\beta}
\end{aligned}$$

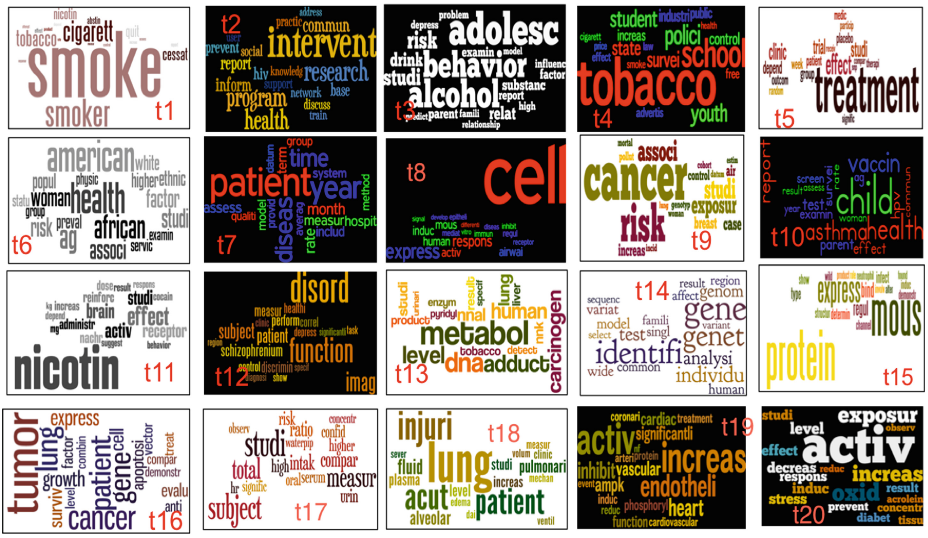

t1 smoke cigarett tobacco nicotin abstin quit cessat smoker
t2 intervent research program health commun practic address user prevent social report hiv knowledg support inform network base discuss train
t3 adolesc behavior alcohol risk drink studi problem depress examin model influenc factor substanc report high predict parent famili relat relationship
t4 tobacco school polici student industri public health cigarett increas control price state law effect smoke survei free advertis youth
t5 treatment medic particip placebo clinic trial studi depend week patient effect therapi outcom group random signific
t6 american white popul physic higher ethnic statu woman health factor group risk preval studi african examin ag associ servic
t7 patient year time group datum term system method assess qualiti model provid diseas averag month measur hospit rate includ
t8 cell signal develop epithel induc mous differenti diseas inhibit mediat vitro immun regul human respons receptor express activ airwai
t9 cancer associ mortal pollut cohort estim control datum air studi lung genotyp woman exposur breast case risk increas incid
t10 report vaccin screen ag result assess survei rate year test examin child hpv commun woman asthma health parent effect
t11 nicotin dose result respons reinforc studi cocain kg increas brain depend effect mg administr activ nachr suggest receptor behavior
t12 disord measur healthi clinic perform correl subject patient depress significantli task region schizophrenium function control discrimin specif diagnosi show imag
t13 studi urinari result specif lung liver enzym product pyridyl nnal human metabol nnk carcinogen level tobacco detect dna adduct
t14 result region sequenc affect genom variat gene model famili variant select test singl genet identifi analysi wide common individu human
t15 show type wild product role neutrophil infect found induc express bind involv alter demonstr structur determin regul channel mous protein
t16 tumor express lung growth factor combin patient gene cell apoptosi vector treat compar demonstr surviv level cancer evalu anti
t17 observ risk ratio concentr confid studi waterpip higher total high intak compar oral serum measur hr signific urin subject
t18 injuri lung measur volum clinic sever fluid studi pulmonari plasma increas mechan acut level edema patient alveolar dai ventil
t19 coronari cardiac treatment activ significantli nf arteri protein increas inhibit vascular event ampk endotheli induc reduc phosphoryl heart function cardiovascular
t20 studi exposur level observ effect activ decreas reduc respons increas induc oxid result stress acrolein prevent concentr diabet tissu

Fig. 1. Word cloud for top words of 20 topics

This model can be understood as a two-stage stochastic process. An author is represented by a probability distribution over topics, and each topic is represented as probability distributions over words.

3.2 Data Gathering and Preprocessing

We obtained all MedLINE citations published by the principle investigators (PIs, 133 in total) of TRS grants funded by the CTP through Tobacco Regulatory Science Research Program (TRSP) (http://prevention.nih.gov/tobacco/portfolio.aspx). Since each article can have multiple authors, the author set considered here are PIs (can appear in any place in the paper) plus the last author of the paper. The final author set includes 2,740 authors. The document set includes those MEDLINE citations with abstract available, resulting in 7460 abstracts.

For each document, we remove stop words using a stop word list available at Mallet software package. We further filter words based on Term Frequency-Inverse Document Frequency (TF-IDF), where words with high document frequencies and relatively insignificant for single document are removed. We then stem the words by applying the potter stemmer [citation] and words with occurrence lower than 2 are discarded.

3.3 Author Topic Modeling Experiment

We ran the author topic modeling developed by [11] on it for 50 iterations. Topic number T is selected as 20. The hyperparameters α and β are fixed as 50/T and 0.01 respectively.

4 Results and Analysis

4.1 Topic Interpretations

Figure 2 shows the ordered proportion of the 20 topics and Fig. 1 shows the word cloud of the top 20 words for each topic. In order to find out what each topic is focused on, we assign each topic a name based on the top 20 words.

We can see that the 20 topics have comparatively balanced distributions ranging from 0.034 to 0.071. One thing worth noting is that some of the topics may be somewhat irrelevant to TRS.

For those relevant to tobacco research, the topics derived have a broad diversity as discussed in the following:

The top 20 words in the most prevalence topic (T1) include *Smoke, cigarette, cessation, abstinence, control, measure*. T2 which ranks 2nd contains words like *intervention, health, program, network, base, train, social, prevent, address, support and community*. T3 focuses on adolescent-related topics, including *alcohol*, *family relationship* and *behavior*. T4 is similar to T3, but emphasizing more on social elements, including *school*, *law*, *industry* and so on. Those topics suggest policy-making

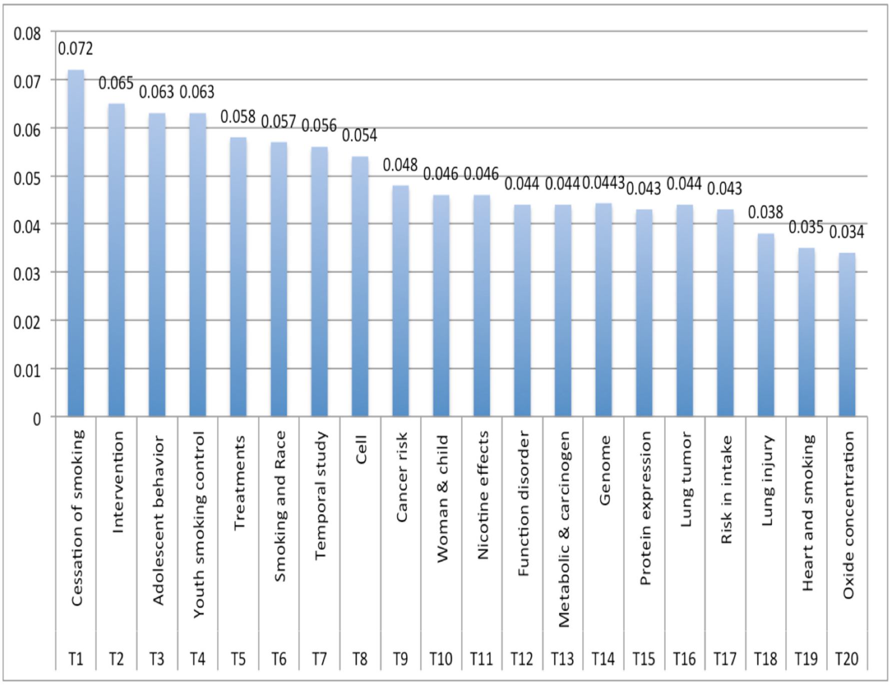


Fig. 2. Topic Proportion

and social studies such as preventing teenager-smoking related research are one of the major trends in TRS research. *Treatment* is most dominant word in T5. Among them, most of words are quite relevant to this keyword. These words suggest that research on clinical practice of smoking related diseases is also tackled by TRS researchers. T6 evidently clusters research on ethnic, gender, age and surveys of smoking revealed by words *American, African, white, group, population, ethnic, woman* and *age*. In contrast, T7 talks about temporal study of smoking-related diseases since temporal words like *time, year, month* and clinical words like *assess, measure, average, quantity, disease* and *datum* are seen there with a good proportion.

Topics from T8 to T20 are all related to direct clinical studies because from now on, we can see there are quite a few domain-specific terms among each topic and understandably, they occupy fewer proportions due to the domain constraints. But the fewer portions do not mean that they are less important for the modeling. On the contrary, they can show how discriminative the author topic modeling can be. For example, the word *cell* is the dominant word in T8. Surrounding it are *words mouse, receptor, express, airway pressure, vitro, response, inhibition, epithelial* and *mediation*. Therefore, this topic talks more about experiments on the influence of smoking on cells. Their proportions are relatively smaller. T9 is obviously discussing relationships between smoking and cancers where *cancer, risk, association, control cohort, air, lung* and *genotype* are the prime terms. Furthermore, *pollution, exposure, woman* and *breast* also suggest that the indirect influence of smoking is included in this topic. As seen, the

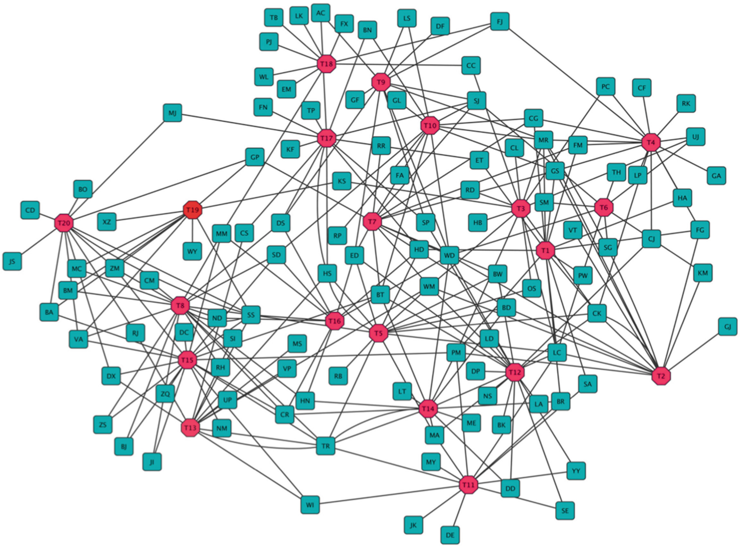


Fig. 3. Author-topic network

core of T10 is *child* with smoking related terms *asthma*, *screen*, *vaccine* and *HPV*. As we mentioned above, all of those topics are more specialized. Without domain knowledge, it can be hard to understand why *HPV* is related to smoking. In fact, according to Troy et al. [12], a case-control study of childhood passive smoke exposure (CPSE) is with human papillomavirus (HPV) infection. *Nicotine* in T11 has the highest proportion, as much as 5 %. It is not hard to imagine that this topic should mainly discuss nicotine and its effects. Words such as *cocaine, brain, response, behavior, kg, mg, reinforce* and *nach* prove this. T12 seems to mainly study the disorder brought by smoking and their correlations. It is composed of words including *disorder, function, schizophrenium, depression, correlation, discrimination* and so on. T13 comprises of a couple of rarely seen terms, such as abbreviations, DNA, NNAL (urinary total 4-(methylnitrosamino)-1-(3-pyridyl)-1-butanonol, which level can be affected by smoking) and nnk (4-(Methylnitrosamino)-1-(3-pyridyl)-1-butanone, one of the most prevalent and procarcinogenic compounds in tobacco), organic chemical elements, pyridyl and enzyme cancer terms, carcinogen, adduction, body and function terms, lung, liver, urinary, metabolic and so on. Among them, metabolic is the leading term unifying all of them. The majority of these topics are related to the harmful and potentially harmful constituents of tobacco products listed as one of the ten interest areas of TRS that have been highlighted by the FDA. *Gene, genetic, genome sequence, individual, variant* and *identify* in T14 show that this research topic on tobacco is from the genetic perspective while *protein, mouse, regulation, binding, express* and so on in T15 more from regulation and binding mechanism of the protein. T16 is also about cancer. However, different from T9, it focuses on lung cancer and treatment. The corresponding cell apoptosis can be indicated from words like *survival, anti, treat, apoptosis* and so on. T17 seems to mainly be related to the medical absorption since we can see words like *intake, concentration, ratio, oral, serum, urinary, waterpipe* and *others* to name a few. T18 also talks about lung, but it is not about lung cancer. Instead, it is more about the general aspects of lung injury since ventilation, plasma, injury, acute and edema are there. Although smoking affects lung so much, T19 tells us that heart diseases are quite related as well where *heart, cardiovascular, cardiac, vascular, endothelial, phosphoric, artery,* and *coronary* are high frequent terms. According to Wheat et al. [13], inhalation of tobacco increases apoptosis and suppresses the VEGF-induced phosphorylation of Akt and endothelial nitric oxide synthases in the aorta. The last one, T20, looks like associating smoke and diabetes through similar mechanism in T19. Acrolein, an element rich in tobacco, is the main element, which prevents the nitric oxide, to lead to smoking-caused diseases. All those 20 topics, we can say that most of them have a good alignment with TRS. This shows that author topic modeling is capable of modeling the topic distributions of the collection.

4.2 Author-Topic Relations

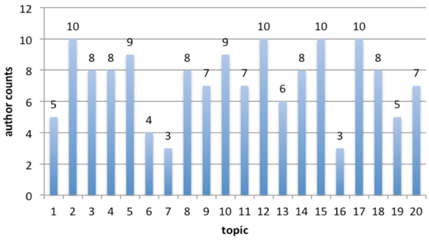


Fig. 4. Author counts in topics maximum

Now, if we turn to the correspondence of top authors (if the author has more than 0.01 portion of articles in that topic, he would be counted as a top author) and topics in Fig. 3, more interesting patterns can be found. Figure 3 is a network with each topic as the hub (the red octagons) and authors form nearest neighbors of each topic if their research involves that topic (the green plate). For better visualization, we use initials for authors to represent them. The correspondence of initials and full names is seen in the supplement. Based on this network, it is found that the top 5 authors in each topic are the prime principle investigators in corresponding topics. For example, Hatsukami D, Cummings K and Eissenberg T, who rank top 3 in T1, are all senior tobacco researchers who mainly focus on tobacco addiction characterization, reduction and/or treatment. Meanwhile, as shown in the network, there are connections between topics. That means that many authors' research areas cover more than one topic.

At first glance, Fig. 4 is similar to Fig. 3, aiming at showing how many authors appear in one topic. But the main goal of Fig. 3 lies in telling us who are top authors in a topic while Fig. 4 tells some topic is the most studied one for some author. For example, the 3 counts for T7 in Fig. 4 indicates that there are three authors whose highest portion are in T7 while the 14 nearest neighbors around T7 in Fig. 3 show that 14 authors have portions larger than 0.01 in their research for topic 7. Figure 4 shows that T2, T12, T15 and T17 are the most studies topics since for each of them, 10 authors published large number of articles on them. This trend does not align with that of topic proportion. To a large degree, we can say that the topic proportion shows that how many researchers are studying what topics while authors counts reflected in Fig. 4 show that which topic has been intensely studied by a few researchers. If we count T5 and T9 (there are 9 authors respectively), the data suggests that tobacco prevention and treatment are popular topics among those researchers.

Another interesting thing is to look at co-occurrence of authors among multiple topics (for simplicity, we only consider two). It can reflect two aspects, one on the closeness of two topics (the two or more can be subtopics of a big topic) and the other on interactions of two topics (they may not be related but depend on each other)

It is found that T15 and T8 co-occur together 10 times, ranking the highest. It means that 10 authors study both topics. Both topics involve *genetic expressions*, *cell*, and *protein*. The combination of T16 and T8 follows closely where topic 16 is about *lung tumor* study from *gene* and *cell* level. The topic dependence relation can be illustrated by the large number of topics co-occurring with T2 (*intervention*). This topic is not really funded by the FDA's CTP, so why do they have such a high proportion of research (0.065)? If we look at other topics which investigators focus on in addition to T2, we can discover clues. Three topics occurring quite commonly with T2 are T1, T3 and T4 (4 times respectively). These four topics are about smoking cessation, vulnerable populations, and youth initiation and access, all of which are TRS priority areas. The link between *smoking cessation* and *intervention* is interesting, as interventions focusing on cessation are specifically mentioned as not a fundable TRS area. Investigators with this topic pair, which is common in tobacco control research in general, may be looking at other related topics that do fall under the TRS scope, such as *nicotine reduction*, *consumer perception* (of certain products as a cessation aid) and *effective communication strategies*. In addition, T7 (*temporal study*) co-occurs with T2 three times as well. This connection between *temporal study* and *intervention* would be a necessary one, as intervention research requires studies across time.

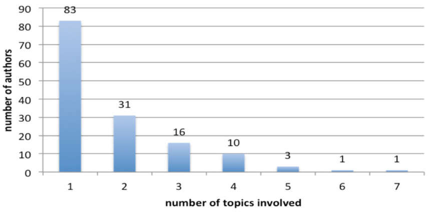


Fig. 5. Author-topic involvement

4.3 Topic Clusters Based on Authors

If we look at authors and topics they are assigned, we can see both extremes. Figure 5 shows that 83 top authors are in fact focus on only one topic. A few authors have high topic involvements. The highest one is *Williams D* who studies 7 topics. The next four are *Srivastava S* (6 topics), *Glantz S* (5 topics), *Baker T* (5 topics) and *Elashoff D* (5 topics) respectively. *Williams D*, as the most diverse researcher, is in fact a leading social and behavior scientist focusing on public health [14]. His research has enhanced the understanding of the complex ways in which race, racial discrimination, socioeconomic status and religious involvement can affect physical and mental health. His topics in the tobacco regulations cross from intervention study, health and race, gender, age, functional disorder and genetic analysis and so on. *Glantz S* is American Legacy Foundation Distinguished Professor of Tobacco Control at the University of California – San Francisco whose research focuses on the health effects of tobacco smoking and who is active in the nonsmokers' rights movement and has advocated for public health polices to reduce smoking. His research topics include T1, T2, T4, T7 and T10, which quite match his research focus.

Baker T is involved in T7, T10, T12, T14 and T15 while *Elashoff D* in topic T5, T7, T9, T12 and T16. They have two overlapping both with T7 and T12 assigned to them. Both of them seem to study topics related to treatment of smoking related diseases. What *Elashoff D* is studying is more cancer related. The topics both of them share are more general aspects like temporal study, function disorder and genetic tests. The remaining topics *Baker T* has, like T10, T14 and T15 involve smoking cessation, intervention, influences on children and protein binding and regulations. On the other hand, *Elashoff D*'s remaining topics including T5, T9 and T16 are all either cancer-related or organ-injury relevant. In *Baker T*'s webpage [15], it states that *Baker T* concentrates on tobacco-dependence treatment and outcomes. He and his team are not only looking at smoking cessation, but also determine how quitting affects the person's physical health, mental health, quality of life and social interactions. Then, *Elashoff D*'s research include statistical analysis of high-throughput microarray, biomarker discovery and validation studies. Meanwhile, he has extensive working on cancer related projects with collaborations in oral, lung, prostate, breast and skin cancers [16]. It seems that those descriptions confirm what we have found from those topics.

As mentioned before, topics discovered are not necessarily all primarily about tobacco and nicotine in this work. Instead, it focuses on finding the interactions between authors, topics and words and what trends can be traced under the frame of TRS. Observing along this thought, we found connections between tobacco and other related topics unique to TRS research. For instance, *Srivastava S*, a project lead on a TRS center grant, is not primarily a tobacco researcher. Instead, he is faculty in an environmental cardiology department. His topic profile includes T6, T8, T13, T15 and T16. From his webpage, we found that his research priority is toxicity, which can explain the connections of the 6 topics: all of them are less or more related to his priority. It is also a topic area prominently featured in the FDA's TRS priority and interest areas.

On other extreme, there are a few PIs who are only assigned one topic. One of them is *Delnevo C* whose topic is T6, which is about ethnic, gender and age related study of smoking. In the website, it says that his research interests are clinical prevention services, tobacco control and survey research methods. Another one is *Donny E* whose topic is T11, which is about the nicotine effects. In his webpage, it says that nicotine reinforcement, regulation of tobacco and implications for health are his primary research interests. Likewise, *Farrelly M* is a leading expert in tobacco control and policy interventions, for youth in particular. The only topic assigned to him is T4, exactly matching his interests.

5 Discussion and Conclusion

5.1 Summary

In this work, we employ author topic modeling to model principle investigators on tobacco and their topics of research based on PubMED literatures. Author topic modeling has been shown to be an effective approach in modeling corpus of computer sciences as well as more general ones, like publically available emails, collections of diverse research articles. No research is done in modeling a constraint domain like tobacco regulations. The results show that this approach can efficiently cluster collections of articles into discriminative categories without any supervision. More interestingly, it can associate topics to authors in a high accuracy. This indicates that we may incorporate author topic modeling into author identification systems to infer the identity of an author of articles using topics generated by the model. The relevance of this analysis to TRS is at least twofold. First, this analysis is a ‘proof of concept’ that can be beneficial assess the change over time in TRS as new projects are funded and collaborative science in this area changes. The results can thus we used to assess the extent to which new research reflects the funding priorities of the FDA. Second, author topic modeling outcomes can be used by investigators to assess who is conducting research in a particular research domain in order to foster collaborative science. By fostering collaborative science in TRS, it becomes possible to speed advances in that science by fostering communication between scientists that can avoiding un-needed duplication and impact decision-making on new science that can benefit regulatory decision-making.

5.2 Limitations and Future Work

One limitation for this approach is that author topic modeling assumes that the topic distribution of each word in one document is only associated with one of the known authors, thus correlations of authors cannot be reflected from words of the same document and instead, must be found across multiple documents, which have the same authors. For large amount of corpus, this may not be a big problem. Nonetheless, this limitation can be overcome if we introduce the topic-author associations as multiple to multiple. As mentioned above, one of the major limitations of our work is the one to one author-word correspondence. Hence, in our future study, we will extend author topic modeling into group author topic modeling. In addition, considering that research topics may change every few years even for the same investigators, it would therefore be reasonable to model temporal changes. One more extension can be that we may build a predictive model based on author topic modeling so that we can assign authors to unknown articles or we can predict what main topics an unknown article is about.

Acknowledgment. This study was made possible by National Science Foundation ABI:0845523, National Institute of Health R01LM009959A1 and R01GM102283A1.

Supplements: Correspondance Author Full Name and Short Name

short	full name	short	full name	short	full name	short	full name
GA	Goldstein, Adam	DE	Donny, Eric	BM	Boehm, Manfred	TR	Turesky, Robert
SA	Sved, Alan	KF	Kamangar, Farin	PM	Picciotto, Marina	MR	Mermelstein,
VA	Vliet, Albert	DF	Dominici, Francesca	EM	Eisner, Mark	BR	Borland, Ron
MA	Mukhin, Alexey	CF	Chaloupka, Frank	WM	Wolfson, Mark	CR	Crystal, Ronald
FA	Ferketich, Amy	GF	Gilliland, Frank	PM	Pentz, Mary	RR	Robertson, Rose
HA	Hyland, Andrew	FG	Fong, Geoffrey	WM	Wewers, Mary	MR	Malone, Ruth
LA	Lees, Andrew	SG	Singh, Gopal	FM	Farrelly, Matthew	DS	Dawsey, Sanford
SA	Strasser, Andrew	CG	Connolly, Gregory	KM	Kreuter, Matthew	SS	Srivastava,
MA	Malhotra, Anil	RH	Reinecke, Hans	NM	Nibert, Max	HS	Heil, Sarah
BA	Bhatnagar, Aruni	TH	Taylor, Herman	PM	Piper, Megan	SS	Srivastava,
TB	Thompson, B	JI	Jaspers, Ilona	CM	Caligiuri, Michael	KS	Kinlay, Scott
HB	Halpern-Felsher, Bonnie	SI	Stepanov, Irina	FM	Fiore, Michael	MS	Murphy, Sharon
RB	Rounsaville, Bruce	WI	Wainer, Irving	MM	Matthay, Michael	SS	Sharma, Sherven
LC	Latkin, Carl	BJ	Blalock, J	WM	Weaver, Michael	SS	Sigmon, Stacey
CC	Calfee, Carolyn	FJ	Frank, James	ZM	Zou, Ming-Hui	GS	Glantz, Stanton
LC	Lerman, Caryn	CJ	Chriqui, Jamie	WM	Wang, Mingyao	LS	London, Stepha-
MC	Murry, Charles	PJ	Pittet, Jean-FranÃ§ois	SM	Siahpush, Mohammad	OS	O'Malley, Ste-
PC	Perry, Cheryl	FJ	Forster, Jean	BN	Benowitz, Neal	HS	Hecht, Stephen
CC	Chou, Chih-Ping	SJ	Smith, Jennifer	FN	Freedman, Neal	HS	Higgins, Stephen
AC	Abnet, Christian	UJ	Unger, Jennifer	HN	Hackett, Neil	NS	Nelsen, Stephen
DC	Doerschuk, Claire	FJ	Freudenheim, Jo	BN	Brewer, Noel	CS	Carmella, Steven
SD	Stein, Dan	MJ	Morris, John	BO	Barski, Oleg	DS	Dubinett, Steven
CD	Conklin, Daniel	RJ	Richie, John	DP	Derosse, Pamela	ZS	Ziegler, Steven
LD	Langleben, Daniel	SJ	Samet, Jonathan	LP	Ling, Pamela	KS	Krishnan-Sarin,
RD	Romer, Daniel	CJ	Carroll, Joseph	RP	Reiter, Paul	JS	Jordt, Sven-Eric
ND	Nguyen, Dao	GJ	Guydish, Joseph	GP	Ganz, Peter	ET	Eissenberg,
HD	Hoon, Dave	MJ	Muscat, Joshua	SP	Shields, Peter	VT	Valente, Thomas
ED	Elashoff, David	CK	Cummings, K	VP	Villalta, Peter	BT	Baker, Timothy
SD	Schrump, David	BK	Burdick, Katherine	SP	Szeszko, Philip	LT	Lencz, Todd
WD	Williams, David	CK	Carroll, Kathleen	TP	Taylor, Philip	PW	Pickworth,
WD	Wong, David	LK	Liu, Kathleen	UP	Upadhyaya, Pramod	BW	Bailey, William
DD	Drayna, Dennis	JK	Jozwiak, Krzysztof	ZQ	Zhang, Qing-Yu	FX	Fang, Xiaohui
HD	Hatsukami, Dorothy	RK	Ribisl, Kurt	TR	Tyndale, Rachel	DX	Ding, Xinxin
BD	Bell, Douglas	CL	Chassin, Laurie	MR	McConnell, Rob	MY	Mineur, Yann
SE	Stein, Elliot	WL	Ware, Lorraine	BR	Balster, Robert	YY	Yang, Yihong
ME	Mongodin, Emmanuel	GL	Gerald, Lynn	TR	Tarran, Robert	WY	Wu, Yong
						XZ	Xie, Zhonglin

References

1. Rosen-Zvi, M., et al.: The author-topic model for authors and documents. In: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence. AUAI Press (2004)
2. Dumais, S.T.: Latent semantic analysis. Annu. Rev. Inf. Sci. Technol. **38**(1), 188–230 (2005)
3. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57 (1999)
4. Turney, P.D., Pantel, P.: From frequency to meaning: Vector space models of semantics. J. Artif. Intell. Res. **37**(1), 141–188 (2010)
5. McCallum, A.: Multi-label text classification with a mixture model trained by EM. In: AAAI'99 Workshop on Text Learning (1999)
6. McCallum, A., Mann, G., Mimno, D.: Bibliometric impact measures leveraging topic analysis. In: Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL'06. IEEE (2006)

7. Steyvers, M., et al.: Probabilistic author-topic models for information discovery. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2004)
8. McCallum, A., Corrada-Emmanuel, A., Wang, X.: The author-recipient-topic model for topic and role discovery in social networks: Experiments with enron and academic email (2005)
9. Bhattacharya, I., Getoor, L.: A latent dirichlet model for unsupervised entity resolution (2005)
10. Newman12, D., Karimi, S., Cavedon, L.: Topic Models to Interpret MeSH–MEDLINE's Medical Subject Headings
11. Mark Steyvers, T.G.: Matlab Topic Modeling Toolbox 1.4 (2014). http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm. cited 7 Oct 2013
12. Troy, J.D., et al.: Childhood passive smoke exposure is associated with adult head and neck cancer. Cancer Epidemiol. 37(4), 417–423 (2013)
13. Wheat, L.A., et al.: Acrolein Inhalation Prevents Vascular Endothelial Growth Factor-Induced Mobilization of Flk-1 +/Sca-1 + Cells in Mice. Arterioscler. Thromb. Vasc. Biol. **31**(7), 1598–1606 (2011)
14. David, H., Williams, R.: School of Public Health (2011). http://www.hsph.harvard.edu/david-williams/. cited 7 Oct 2013
15. Center for Tobacco Research and Intervention, U.o.W. http://www.ctri.wisc.edu/News.Center/News.Center_bio_tim_baker.html. cited 1 Oct 2013
16. Department of Biostatistics, U. http://www.biostat.ucla.edu/Directory/Delashoff. cited 1 Oct 2013

Mining Severe Drug-Drug Interaction Adverse Events Using Semantic Web Technologies: A Case Study

Guoqian Jiang(✉), Hongfang Liu, Harold R. Solbrig,
and Christopher G. Chute

Department of Health Sciences Research, Mayo Clinic, Rochester, MN, USA
{jiang.guoqian, liu.hongfang, solbrig.harold,
chute}@mayo.edu

Abstract. Drug-drug interactions (DDIs) are a major contributing factor for unexpected adverse drug events (ADEs). However, few of knowledge resources cover the severity information of ADEs that is critical for prioritizing the medical need. The objective of the study is to develop and evaluate a Semantic Web-based approach for mining severe DDI-induced ADEs. We utilized a normalized FDA Adverse Event Report System (AERS) dataset and performed a case study of three frequently prescribed cardiovascular drugs: Warfarin, Clopidogrel and Simvastatin. We extracted putative DDI-ADE pairs and their associated outcome codes. We developed a pipeline to validate the associations using ADE datasets from SIDER and PharmGKB. We also performed a cross validation using electronic medical records (EMR) data. We leveraged the Common Terminology Criteria for Adverse Event (CTCAE) grading system and classified the DDI-induced ADEs into the CTCAE in the Web Ontology Language (OWL). We identified and validated 601 DDI-ADE pairs for the three drugs using the validation pipeline, of which 61 pairs are in Grade 5, 56 pairs in Grade 4 and 484 pairs in Grade 3. Among 601 pairs, the signals of 59 DDI-ADE pairs were identified from the EMR data. The approach developed could be generalized to detect the signals of putative severe ADEs induced by DDIs in other drug domains and would be useful for supporting translational and pharmacovigilance study of severe ADEs.

Keywords: Drug-drug interaction · Adverse drug events · Adverse event report system (AERS) · Severity · Semantic web technologies

1 Introduction

Drug-drug interactions (DDIs) are a major contributing factor for unexpected adverse drug events (ADEs) [1]. A semantically coded knowledge base of DDI-induced ADEs with severity information is critical for clinical decision support systems and translational research applications. In particular, there is emerging interest in investigating genetic susceptibility of DDI-induced ADEs and developing genetic tests to identify all those at risk of ADEs prior to prescribing potentially dangerous medication [2, 3], in which the severity information is essential for prioritizing the medical need to evaluate

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 628–638, 2014.
DOI: 10.1007/978-3-319-13186-3_56

the potential impact of pharmacogenomics information in reducing ADEs [4]. However, few of knowledge resources cover severity information of ADEs.

While recognizing, explaining and ultimately predicating DDIs constitute a huge challenge for medicine and public health, informatics-based approaches are increasingly used in dealing with the challenge [5]. Semantic web technologies provide a scalable framework for data standardization and data integration from heterogeneous resources. For instance, Samwald et al. [6] developed a Semantic Web-based knowledge base for query answering and decision support in clinical pharmacogenetics, in which three dataset components are integrated. In our previous and ongoing study, we developed a standardized knowledge base of ADEs known as ADEpedia (http://adepedia.org) leveraging Semantic Web technologies [7]. The ADEpedia is intended to integrate existing known ADE knowledge for drug safety surveillance from disparate resources such as FDA Structured Product Labeling (SPL) [7], FDA Adverse Event Reporting System (AERS) [8], and the Unified Medical Language System (UMLS) [9].

The objective of the study is to develop and evaluate a Semantic Web-based approach for mining severe DDI-induced ADEs. We utilized a normalized FDA AERS dataset and performed a case study of three frequently prescribed cardiovascular drugs: Warfarin, Clopidogrel and Simvastatin. We extracted putative DDI-ADE pairs and their associated outcome codes. We developed a validation pipeline to validate the associations using ADE datasets from SIDER and PharmGKB. We also performed a cross validation using electronic medical records (EMR) data. We leveraged the Common Terminology Criteria for Adverse Event (CTCAE) grading system and classified the DDI-induced ADEs into the CTCAE in the Web Ontology Language (OWL).

2 Background

2.1 FDA Adverse Event Reporting System (AERS)

FDA AERS is a database that provides information on adverse event and medication error reports submitted to FDA [10]. By the definition of FDA, the “serious” means that one or more of the following outcomes were documented in the report: death (DE), hospitalization (HO), life threatening (LT), disability (DS), congenital anomaly (CA) and/or other (OT) serious outcome. In our previous study, we produced a normalized AERS dataset known as AERS-DM [11]. The dataset contains 4,639,613 unique putative Drug-ADE pairs in which the drugs are represented by RxNorm [12] codes and the putative ADEs are represented by MedDRA [13] codes. The data set also contains the unique ID number (known as ISR) for each corresponding AERS report, which is a primary link field between the AERS data file. We used the ISR field to identify the outcome codes of each AERS report. Table 1 shows the outcome code definitions in AERS database.

2.2 Common Terminology Criteria for Adverse Event (CTCAE)

CTCAE is a widely accepted, standard grading scale for adverse events throughout the oncology research community [14]. The current released version is CTCAE 4.0.

This version contains 764 AE terms and 26 "Other, specify" options for reporting text terms not listed in CTCAE. Each AE term is associated with a 5-point severity scale. The AE terms are grouped by MedDRA Primary SOC classes. In the CTCAE, "Grade" refers to the severity of the adverse event (AE). The CTCAE displays Grades 1 through 5 with unique clinical descriptions of severity for each AE based on a general guideline. Table 2 shows the grade definitions in the CTCAE grading system.

Table 1. Outcome code definitions in AERS database

Outcome code	Definition
DE	Death
LT	Life-Threatening
HO	Hospitalization - Initial or Prolonged
DS	Disability
CA	Congenital Anomaly
RI	Required Intervention to Prevent Permanent Impairment/Damage
OT	Other

Table 2. Grade definitions in the CTCAE grading system

Grade	Definition
Grade 1	Mild; asymptomatic or mild symptoms; clinical or diagnostic observations only; intervention not indicated
Grade 2	Moderate; minimal, local or noninvasive intervention indicated; limiting age-appropriate instrumental ADL*
Grade 3	Severe or medically significant but not immediately life-threatening; hospitalization or prolongation of hospitalization indicated; disabling; limiting self care ADL**
Grade 4	Life-threatening consequences; urgent intervention indicated
Grade 5	Death related to AE

Note: Activities of Daily Living (ADL)

* Instrumental ADL refer to preparing meals, shopping for groceries or clothes, using the telephone, managing money, etc.

** Self care ADL refer to bathing, dressing and undressing, feeding self, using the toilet, taking medications, and not bedridden.

2.3 ADE Datasets

SIDER (**SID**e **E**ffect **R**esource) is a public, computer-readable side effect resource that contains information on marketed medicines and their recorded adverse drug reactions [15]. The information is extracted from public documents and package inserts, in particular, from the US FDA Structured Product Labels (SPLs). The current version was released on October 17, 2012.

PharmGKB DDI-ADE Dataset is a database of DDI side effects based on FDA AERS reporting data [16], in which the confounding factors for prediction of the side effects are corrected through leveraging covariates in observational clinical data [17].

2.4 Semantic Web Technologies

The World Wide Web consortium (W3C) is the main standards body for the World Wide Web [18]. The goal of the W3C is to develop interoperable technologies and tools as well as specifications and guidelines to lead the web to its full potential. The resource description framework (RDF), web ontology language (OWL), and SPARQL (a recursive acronym for **S**PARQL **P**rotocol and **R**DF **Q**uery **L**anguage) specifications have all achieved the level of W3C recommendations, and are becoming generally accepted and widely used. RDF is a model of directed, labeled graphs that use a set of triples. Each triple is modeled in the form of subject, predicate and object. SPARQL is a standard query language for RDF graphs. OWL is a standard ontology language used for ontology modeling.

3 Methods

In this study, we utilized a normalized AERS dataset known as AERS-DM that was produced in a previous study [11]. The dataset contains 4,639,613 unique putative Drug-ADE pairs in which the drugs are represented by RxNorm codes and the putative ADEs are represented by MedDRA codes.

Figure 1 shows the system architecture of our approach. We first extracted a subset of putative DDI-ADE pairs (in which only two drugs are listed on a report) with their associated outcome codes from original AERS-DM dataset.

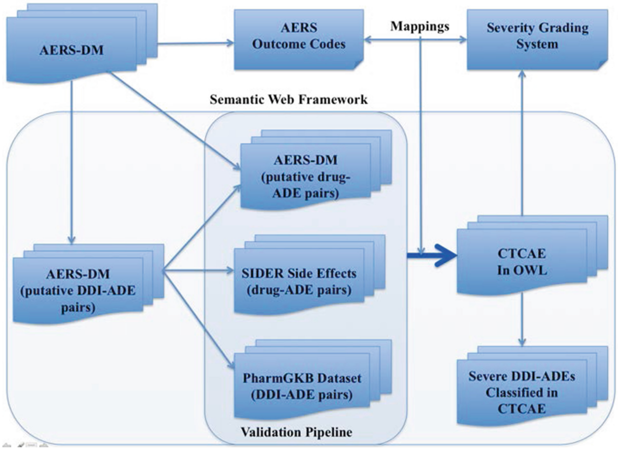

AERS-DM
AERS Outcome Codes
Mappings
Severity Grading System
Semantic Web Framework
AERS-DM (putative drug-ADE pairs)
AERS-DM (putative DDI-ADE pairs)
SIDER Side Effects (drug-ADE pairs)
CTCAE In OWL
PharmGKB Dataset (DDI-ADE pairs)
Severe DDI-ADEs Classified in CTCAE
Validation Pipeline

Fig. 1. System architecture.

Second, we developed a validation pipeline that comprises three datasets. The first dataset is a subset of original AERS-DM in which only one drug is listed on a report. This dataset was used to build a knowledge base of severe ADEs in a previous study. The second one is the SIDER 2 dataset. Table 3 shows a list of drug-ADE pair examples from the dataset, in which drug names are coded in STICH ID (http://stitch.embl.de) and ADE names are coded in MedDRA. We excluded the putative DDI-ADE pairs based on the Drug-ADE pairs of the two datasets. The validation would ensure that the reported ADEs could not be explained by a single drug effect. The third one is a PharmGKB dataset that is used as "silver" standard. Table 4 shows a list of DDI-ADE examples from the dataset, in which drug names are coded in STICH ID and ADE names are coded in UMLS Concept Unique Identifiers (CUIs).

Third, we converted all the datasets used in this study into the Semantic Web RDF format and loaded them into an open source RDF store known as 4store [19]. We established a SPARQL endpoint that provides standard query services against the RDF store. And then we developed the extraction and validation algorithms using Java-based Jena ARQ APIs [20].

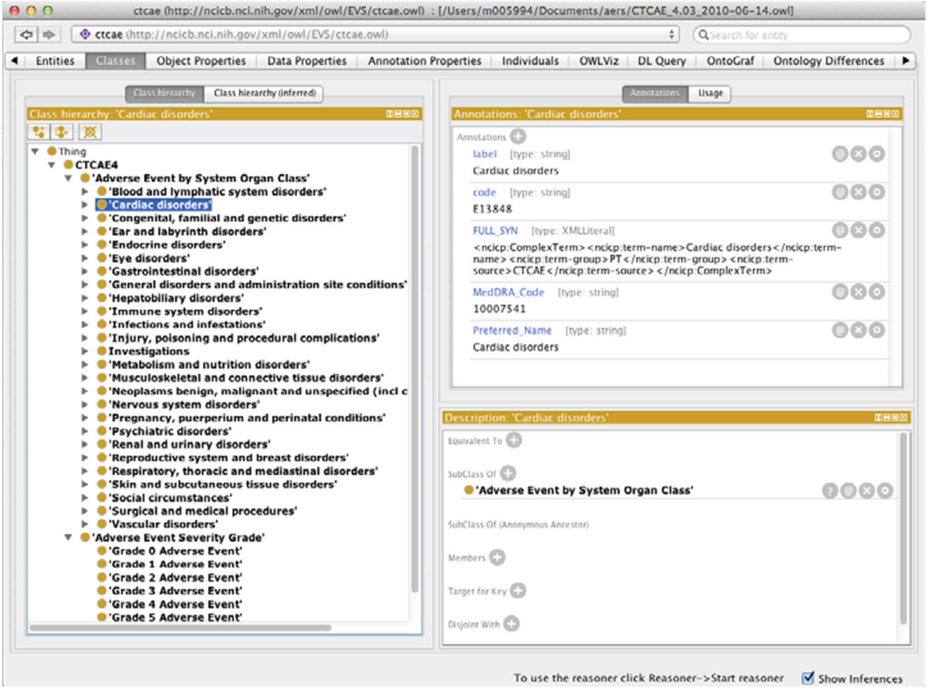
Third, to cross-validate the DDI-induced ADEs, we used the NLP-processed EMR data of a cohort of 138 k patients with health home care provided by Mayo Clinic Rochester where medications and problems have been extracted and normalized to RxNorm codes and the UMLS concepts from the medical records using MedTagger and MedXN (http://www.ohnlp.org/). For each DDI-induced ADE triples (D1, D2, P), we obtained the number of patients who are administrated with any of the two drugs or both (i.e., N(D1), N(D2), and N(D1,D2)) and the number of patients with putative ADEs (i.e., N(D1,P), N(D2,P), and N(D1,D2,P)) after taking the drugs. An occurrence of problem P is considered as putative ADE if it happens within 36 days of drug administration [1, 17] and there is no occurrence of P in the EMR before the drug administration. We then used the following metric to measure the signal enrichment of DDI-induced ADE:

$$Score(D1, D2, P) = log_2\left(\frac{N(D1,D2,P)}{N(D1,D2)}\Big/ \max\left(\frac{N(D1,P)}{N(D1)}, \frac{N(D2,P)}{N(D2)}\right)\right).$$

Finally, we developed the mappings between AERS outcome codes and CTCAE grades and classified validated DDI-ADEs into the CTCAE. We asserted that DE in AERS corresponds to Grade 5 in CTCAE; LT corresponds to Grade 4; the rest of outcome codes (HO, DS, CA, RI and OT) correspond to Grade 3. In this study, we utilized the CTCAE version 4.0 [14] rendered in OWL format. Figure 2 shows a screenshot of a Protégé4 environment displaying the categories and severity grades in CTCAE classification.

4 Results

We were able to extract a set of putative DDI-ADE pairs and their associated outcome codes for the three target drugs: Warfarin, Clopidogrel and Simvastatin from normalized AERS-DM dataset. We then validated the putative DDI-ADE pairs using the


ctcae (http://ncicb.nci.nih.gov/xml/owl/EVS/ctcae.owl) : [/Users/m005994/Documents/aers/CTCAE_4.03_2010-06-14.owl]
ctcae (http://ncicb.nci.nih.gov/xml/owl/EVS/ctcae.owl)
Search for entity
Entities
Classes
Object Properties
Data Properties
Annotation Properties
Individuals
OWLViz
DL Query
OntoGraf
Ontology Differences
Class hierarchy
Class hierarchy (inferred)
Class hierarchy: 'Cardiac disorders'
Thing
CTCAE4
'Adverse Event by System Organ Class'
'Blood and lymphatic system disorders'
'Cardiac disorders'
'Congenital, familial and genetic disorders'
'Ear and labyrinth disorders'
'Endocrine disorders'
'Eye disorders'
'Gastrointestinal disorders'
'General disorders and administration site conditions'
'Hepatobiliary disorders'
'Immune system disorders'
'Infections and infestations'
'Injury, poisoning and procedural complications'
Investigations
'Metabolism and nutrition disorders'
'Musculoskeletal and connective tissue disorders'
'Neoplasms benign, malignant and unspecified (incl c
'Nervous system disorders'
'Pregnancy, puerperium and perinatal conditions'
'Psychiatric disorders'
'Renal and urinary disorders'
'Reproductive system and breast disorders'
'Respiratory, thoracic and mediastinal disorders'
'Skin and subcutaneous tissue disorders'
'Social circumstances'
'Surgical and medical procedures'
'Vascular disorders'
'Adverse Event Severity Grade'
'Grade 0 Adverse Event'
'Grade 1 Adverse Event'
'Grade 2 Adverse Event'
'Grade 3 Adverse Event'
'Grade 4 Adverse Event'
'Grade 5 Adverse Event'
Annotations
Usage
Annotations: 'Cardiac disorders'
Description: 'Cardiac disorders'
Equivalent To
SubClass Of
SubClass Of (Anonymous Ancestor)
Members
Target for Key
Disjoint With
To use the reasoner click Reasoner->Start reasoner
☑ Show Inferences

Annotations

label [type: string]
Cardiac disorders

code [type: string]
E13848

FULL_SYN [type: XMLLiteral]
<ncicp:ComplexTerm><ncicp:term-name>Cardiac disorders</ncicp:term-name><ncicp:term-group>PT</ncicp:term-group><ncicp:term-source>CTCAE</ncicp:term-source></ncicp:ComplexTerm>

MedDRA_Code [type: string]
10007541

Preferred_Name [type: string]
Cardiac disorders

SubClass Of
'Adverse Event by System Organ Class'

Fig. 2. The categories and severity grades of CTCAE classification in a Protégé 4 environment.

Table 3. A list of Drug-ADE examples from SIDER dataset, in which drug names are coded in STICH ID and ADE names are coded in MedDRA.

stitch_idl	stitch_id2	UMLS_concept_id	drug_name	side_effect_name	MedDRA_concept_type	UMLS_concept_id	MedDRA_side_effect_name
−100003914	−39468	C0038454	levobunolol	ccrebrovascu	LLT	C0038454	Cerebrovascular accident
−100003914	−39468	C0038454	levobunolol	ccrebrovascu	PT	C0038454	Cerebrovascular accident
−100003914	−39468	C0015230	levobunolol	rash	LLT	C0015230	Rash
−100003914	−39468	C0015230	levobunolol	rash	PT	C0015230	Rash
−100003914	−39468	C0015230	levobunolol	rash	PT	C0011603	Dermatitis
−100003914	−39468	C0033377	levobunolol	ptosis	LLT	C0033377	Ptosis
−100003914	−39468	C0033377	levobunolol	ptosis	PT	C000S745	Eyelid ptosis
−100003914	−39468	C0033377	levobunolol	ptosis	PT	C0156353	Uterovaginal prolapse
−100003914	−39468	C0030554	levobunolol	paresthesia	LLT	C0030554	Paraesthesia
−100003914	−39468	C0030554	levobunolol	paresthesia	PT	C0030554	Paraesthesia
−100003914	−39468	C0006266	levobunolol	bronchospasr	LLT	C0006266	Bronchospasm
−100003914	−39468	C0006266	levobunolol	bronchospasr	PT	C0006266	Bronchospasm
−100003914	−39468	C114S670	levobunolol	respiratory fa	LLT	C1145670	Respiratory failure
−100003914	−39468	C114S670	levobunolol	respiratory fa	PT	C1145670	Respiratory failure
−100003914	−39468	C0027424	levobunolol	nasal congest	LLT	C0027424	Nasal congestion
−100003914	−39468	C0027424	levobunolol	nasal congest	PT	C0027424	Nasal congestion
−100003914	−39468	C0023380	levobunolol	lethargy	LLT	C0023380	Lethargy
−100003914	−39468	C0023380	levobunolol	lethargy	PT	C0023380	Lethargy
−100003914	−39468	C0947912	levobunolol	myasthenia	LLT	C0947912	Myasthenia
−100003914	−39468	C0947912	levobunolol	myasthenia	PT	C0151786	Muscular weakness

validation pipeline based on three datasets. Table 5 shows the number of validated DDI-ADE pairs for each target drug. In total, 601 pairs were validated. Of them, 61 pairs are classified in Grade 5, 56 pairs in Grade 4 and 484 pairs in Grade 3. Table 6

Table 4. A list of DDI-ADE examples from PharmGKB dataset, in which drug names are coded in STICH ID and ADE names are coded in UMLS CUI.

stitch_idl	stitch_id2	drug1	drug2	event_umls_id	event_name
CID000000085	CID000000206	carnitine	galactose	C0004623	Bacterial infection
CID000000085	CID000000206	carnitine	galactose	C0015967	body temperature increased
CID000000085	CID000000206	carnitine	galactose	C0018932	haematochezia
CID000000085	CID000000206	carnitine	galactose	C0020433	Bilirubinaemia
CID000000085	CID000000206	carnitine	galactose	C0022346	icterus
CID000000085	CID000000206	carnitine	galactose	C0026946	fungal disease
CID000000085	CID000000206	carnitine	galactose	CG030305	pancreatitis
CID000000085	CID000000206	carnitine	galactose	C0040034	thrombocytopenia
CID000000085	CID000000206	carnitine	galactose	C0085605	Hepatic failure
CID000000085	CID000000206	carnitine	galactose	C0151766	Abnormal LFTs
CID000000085	CID000000206	carnitine	galactose	C0243026	sepsis
CID000000085	CID000000271	carnitine	calcium	C0002792	anaphylactic reaction
CID000000085	CID000000271	carnitine	calcium	C0002871	anaemia
CID000000085	CID000000271	carnitine	calcium	C0002962	angina
CID000000085	CID000000271	carnitine	calcium	C0004238	AFIB
CID000000085	CID000000271	carnitine	calcium	C0010054	arteriosclerotic heart disease
CID000000085	CID000000271	carnitine	calcium	C0010200	Cough
CID000000085	CID000000271	carnitine	calcium	C0012833	dizziness
CID000000085	CID000000271	carnitine	calcium	C0013404	Difficulty breathing
CID000000085	CID000000271	carnitine	calcium	C0015802	femur fracture

Table 5. The number of validated DDI-ADE pairs for three drugs.

Drug	Number of DDI-ADE pairs		
	Grade 5	Grade 4	Grade 3
Warfarin	32	11	157
Clopidogrel	17	29	166
Simvastatin	12	16	161
Total	61	56	484

shows a list of validated DDI-ADE pair examples for the drug "Simvastatin", in which, drugs are coded in RxNorm RxCUIs and ADEs are coded in MedDRA codes.

For the cross-validation using the EMR data, we found that, there are 89 drug pairs prescribed concomitantly in 9.5 k patients, accounting for 6.9 % of all patients in the EMR dataset we used. Out of 601 putative DDI-ADE pairs, the signals of 59 (D1, D2, P) pairs were identified. Table 7 shows the detailed statistics of those pairs occurred in no less than five patients.

For integrating the validated DDI-ADE pairs with the CTCAE, we produced an OWL rendering for each pair, asserting the validated DDI-ADEs under AE terms in CTCAE (see Fig. 3 for an example).

5 Discussion

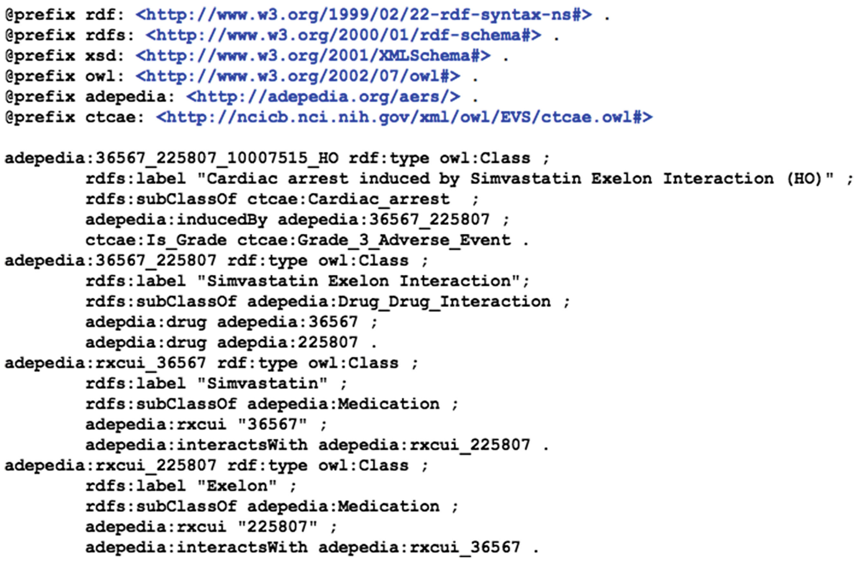
In a previous study, we used a similar Semantic Web-based approach to build a knowledge base of severe ADEs using the FDA AERS reporting data [8]. In this study,

Table 6. A list of validated DDI-ADE pairs for the drug “Simvastatin” classified by CTCAE grades

CTCAE grade	AERS outome code	Drug code by RxCUI	Drug name	Drag code by RxCUI	Drug name	ADE code by MedDRA	ADE name
Grade 5	DE	36567	Simvastatin	1911	Aspirin	10002906	Aortic stenosis
Grade 5	DE	253198	Rosiglitazone maleate	36567	Simvastatin	I0006580	Bundle branch block left
Grade 5	DE	36567	Simvastatin	203160	Losartan Potassium	10007515	Cardiac arrest
Grade 5	DE	36567	Simvastatin	1191	Aspirin	10010071	Coma
Grade 5	DE	253198	Rosiglitazone maleate	36567	Simvastatin	10012689	Diabetic retinopathy
Grade 4	LI	36567	Simvastatin	203029	Tegretol	10002948	Aphasia
Grade 4	LI	36567	Simvastatin	203029	Tegretol	10003119	Arrhythmia
Grade 4	LI	203114	Amiodarone hydrochloride	316675	Simvastatin 80 MG	10006002	Bone pain
Grade 4	LT	36567	Simvastatin	225807	Exelon	10007515	Cardiac arrest
Grade 4	LI	36567	Simvastatin	203029	Tegretol	10012455	Dermatitis exfoliative
Grade 3	DS	36567	Simvastatin	1191	Aspirin	10012455	Denuatitis exfoliative
Grade 3	DS	36567	Simvastatin	190465	Viagra	10018429	Glucose tolerance impaired
Grade 3	DS	36567	Simvastatin	83 367	atorvmtin	10020765	Hypersomnia
Grade 3	DS	36567	Simvastatin	35296	Ramipril	10050296	Intervertebral disc protrusion
Grade 3	HO	317636	Gemfibrozil 600 MG	316675	Simvastatin 80 MG	10000486	Acidosis

Table 7. A list of putative DDI-ADE pairs signaled in the EMR data. D1 - drug1, D2 - drug 2, P - problem, N – number, and Score – enrichment score.

D1 (RxCUI)	D2 (RxCUI)	P (MedDRA)	ADE name	N (D1)	N (D2)	N (D1, D2)	N (D1, P)	N (D2, P)	N(D1, D2,P)	Seore (Dl, D2, P)
1191	36567	10002906	Aortic stenosis	38149	7494	2926	104	34	15	4.991
196503	36567	10038428	Renal disorder	10894	7494	1472	40	56	7	4.550
36567	83367	10028417	Myasthenia gravis	7494	2841	828	42	10	5	4.409
11289	3407	10013887	Dysarthria	6330	1927	641	43	7	6	4.360
1191	36567	10015090	Epistaxis	38149	7494	2926	126	28	9	4.257
25480	36567	10019245	Hearing impaired	4683	7494	280	35	70	5	3.935
174742	36567	10017955	Gastrointestinal haemorrhage	4769	7494	642	54	42	9	3.880
1191	32968	10037423	Pulmonary oedema	38149	1436	1291	142	8	8	3.338
1191	32968	10005191	Blister	38149	1436	1291	135	9	7	3.048
17767	36567	10013971	Dyspnoea exertional	2786	7494	561	62	89	11	2.995
1191	36567	10047924	Wheezing	38149	7494	2926	354	73	27	2.969
261551	36567	10012680	Diabetic neuropathy	1883	7494	329	39	20	5	2.630
1191	32968	10038428	Renal disorder	38149	1436	1291	175	9	6	2.452
1191	32968	10040882	Skin lesion	38149	1436	1291	269	21	16	2.024
1191	32968	10046555	Urinary retention	38149	1436	1291	292	16	11	1.757
1191	32968	10061623	Adverse drug reaction	38149	1436	1291	368	20	15	1.549
36567	58927	10017076	Fracture	7494	3416	318	139	59	6	1.219



```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix adepedia: <http://adepedia.org/aers/> .
@prefix ctcae: <http://ncicb.nci.nih.gov/xml/owl/EVS/ctcae.owl#>

adepedia:36567_225807_10007515_HO rdf:type owl:Class ;
        rdfs:label "Cardiac arrest induced by Simvastatin Exelon Interaction (HO)" ;
        rdfs:subClassOf ctcae:Cardiac_arrest  ;
        adepedia:inducedBy adepedia:36567_225807 ;
        ctcae:Is_Grade ctcae:Grade_3_Adverse_Event .
adepedia:36567_225807 rdf:type owl:Class ;
        rdfs:label "Simvastatin Exelon Interaction";
        rdfs:subClassOf adepedia:Drug_Drug_Interaction ;
        adepdia:drug adepedia:36567 ;
        adepdia:drug adepdia:225807 .
adepedia:rxcui_36567 rdf:type owl:Class ;
        rdfs:label "Simvastatin" ;
        rdfs:subClassOf adepedia:Medication ;
        adepedia:rxcui "36567" ;
        adepedia:interactsWith adepedia:rxcui_225807 .
adepedia:rxcui_225807 rdf:type owl:Class ;
        rdfs:label "Exelon" ;
        rdfs:subClassOf adepedia:Medication ;
        adepedia:rxcui "225807" ;
        adepedia:interactsWith adepedia:rxcui_36567 .
```

Fig. 3. The OWL representation of an example DDI-ADE

we focused on mining the DDI-induced ADEs and their severity information, and configured the validation pipeline differently using a collection of ADE datasets. The standardization of ADE datasets is essential for enabling interoperability and comparability heterogeneous data sources. We used a normalized AERS dataset, in which the drug names are normalized using standard drug ontologies RxNorm and NDF-RT and the ADEs are normalized using MedDRA, whereas the datasets from SIDER and PharmGKB used STITCH compound IDs to code drug names and used UMLS CUIs to code ADEs. Apparently, the solid mappings between RxNorm codes and STITCH IDs would be required in future, which will be part of our research efforts in constructing a standardized drug and pharmacological class network [21].

We also tested the signals of putative DDI-ADE pairs validated by the pipeline using a large EMR data. We were able to detect some strong signals indicated by the enrichment score as illustrated in Table 7. This would potentially provide a very useful tool for the knowledge-driven detection of the DDI-induced ADEs from the EMR, though a rigorous patient chart review with a panel of clinicians would be needed in future to verify the signals to establish the causality of the drug-drug interaction.

For measuring the severity of ADEs, we used the CTCAE severity grading system. We found that the AERS outcome codes used to record serious patient outcomes in the AERS reporting data correspond well to the CTCAE Grades 3 to 5. Semantic web OWL rendering of the DDI-ADE dataset provides seamless integration with the CTCAE itself, enabling a standard infrastructure for automatic classification of ADEs based on the severity conditions specified in the CTCAE.

In summary, we developed a Semantic Web-based approach to mine severe DDI-induced ADEs. The dataset produced in this study will be publicly available from our

ADEpedia website (http://adepedia.org). The approach developed could be generalized to detect the signals from EMR for putative severe ADEs induced by DDIs in other drug domains and would be useful for supporting translational and pharmacovigilance study of severe ADEs.

Acknowledgements. The study is supported in part by the SHARP Area 4: Secondary Use of EHR Data (90TR000201).

References

1. Tatonetti, N.P., Fernald, G.H., Altman, R.B.: A novel signal detection algorithm for identifying hidden drug-drug interactions in adverse event reports. J. Am. Med. Inf. Assoc. JAMIA **19**(1), 79–85 (2012)
2. Daly, A.K.: Pharmacogenomics of adverse drug reactions. Genome Med. **5**(1), 5 (2013)
3. Wang, L., McLeod, H.L., Weinshilboum, R.M.: Genomics and drug response. New Engl. J. Med. **364**(12), 1144–1153 (2011)
4. Phillips, K.A., Veenstra, D.L., Oren, E., Lee, J.K., Sadee, W.: Potential role of pharmacogenomics in reducing adverse drug reactions: a systematic review. JAMA J. Am. Med. Assoc. **286**(18), 2270–2279 (2001)
5. Percha, B., Altman, R.B.: Informatics confronts drug-drug interactions. Trends Pharmacol. Sci. **34**(3), 178–184 (2013)
6. Samwald, M., Freimuth, R., Luciano, J.S., et al.: An RDF/OWL knowledge base for query answering and decision support in clinical pharmacogenetics. Stud. Health Technol. Inf. **192**, 539–542 (2013)
7. Jiang, G., Solbrig, H.R., Chute, C.G.: ADEpedia: a scalable and standardized knowledge base of Adverse Drug Events using semantic web technology. In: AMIA Annual Symposium Proceedings 2011, pp. 607–616 (2011)
8. Jiang, G., Wang, L., Liu, H., Solbrig, H.R., Chute, C.G.: Building a knowledge base of severe adverse drug events based on AERS reporting data using semantic web technologies. Stud. Health Technol. Inf. **192**, 496–500 (2013)
9. Jiang, G, Liu, H.F., Solbrig, H.R., Chute, C.G.: ADEpedia 2.0: integration of normalized adverse drug events (ADEs) knowledge from the UMLS. AMIA Jt. Summits Trans. Sci. Proc. **18**, 100–104 (2013)
10. The FDA AERS. http://www.fda.gov/Drugs/GuidanceComplianceRegulatoryInformation/Surveillance/AdverseDrugEffects/default.htm. cited 4 June 2013
11. Wang, L., Jiang, G., Li, D., Liu, H.: Standardizing drug adverse event reporting data. Stud. Health Technol. Inf. **192**, 1101 (2013)
12. Nelson, S.J., Zeng, K., Kilbourne, J., Powell, T., Moore, R.: Normalized names for clinical drugs: RxNorm at 6 years. J. Am. Med. Inf. Assoc. JAMIA **18**(4), 441–448 (2011)
13. The MedDRA. http://www.meddramsso.com/. cited 16 November 2012
14. The CTCAE v4.0. http://evs.nci.nih.gov/ftp1/CTCAE/About.html. cited 1 June 2013
15. Kuhn, M., Campillos, M., Letunic, I., Jensen, L.J., Bork, P.: A side effect resource to capture phenotypic effects of drugs. Mol. Syst. Biol. **6**, 343 (2010)
16. PharmGKB Dataset. http://www.pharmgkb.org/downloads.jsp. cited 8 April 2013
17. Tatonetti, N.P., Ye, P.P., Daneshjou, R., Altman, R.B.: Data-driven prediction of drug effects and interactions. Sci. Transl. Med. **4**(125), 125 (2012)
18. The World Wide Web Consortium (W3C). http://www.w3.org/. cited 25 May 2013

19. Duke, J.D., Li, X., Grannis, S.J.: Data visualization speeds review of potential adverse drug events in patients on multiple medications. J. Biomed. Inform. **43**(2), 326–331 (2010)
20. Ross, C.J., Visscher, H., Sistonen, J., et al.: The Canadian Pharmacogenomics Network for Drug Safety: a model for safety pharmacology. Thyroid. **20**(7), 681–687 (2010)
21. Zhu, Q., Jiang, G., Wang, L., Chute, C.G.: Standardized drug and pharmacological class network construction. In: ICBO 2013 - Vaccine and Drug Ontology Studies (VDOS-2013) Workshop, Montreal, Qc. Canada (2013)

Cancer Based Pharmacogenomics Network for Drug Repurposing

Liwei Wang[1], Hongfang Liu[2], Christopher G. Chute[2],
and Qian Zhu[3](✉)

[1] Department of Medical Informatics, School of Public Health, Jilin University,
Changchun 130021, China
wlw@jlu.edu.cn

[2] Department of Health Science Research, Mayo Clinic,
Rochester, MN 55901, USA
{liu.hongfang, chute}@mayo.edu

[3] Department of Information Systems,
University of Maryland Baltimore County, Baltimore, MD 21250, USA
qianzhu@umbc.edu

Abstract. Pharmacogenomics (PGx) as an emerging field, is poised to change the way we practice medicine and deliver health care by customizing drug therapies on the basis of each patient's genetic makeup. A large volume of PGx data including information on relationships among drugs, genes, and single nucleotide polymorphisms (SNPs) has been accumulated. Normalized and integrated PGx information could facilitate revelation of hidden relationships among drug treatments, genomic variations, and phenotype traits to better support drug discovery and next generation of treatment. In this study, we constructed a normalized cancer based PGx network (CPN) by integrating cancer orientated PGx information from multiple well known PGx resources including the Pharmacogenomics Knowledge Base (PharmGKB), the FDA Pharmacogenomic Biomarkers in Drug Labeling, and the Catalog of Published Genome-Wide Association Studies. The ultimate goal of the CPN is to provide comprehensive cancer specific PGx information to support oncology related research, including cancer based drug discovery – drug repurposing. We have successfully demonstrated the capability of the CPN for drug repurposing by conducting two case studies.

Keywords: Pharmacogenomics · Cancer · Network · Drug repurposing

1 Introduction

In 2003, the US Food and Drug Administration (FDA) recognized the importance of PGx data for the evaluation of drug safety and efficacy by starting a voluntary data exchange program, which requests that pharmaceutical companies submit genomic data along with their new drug packages. So far, the FDA has documented PGx information for more than 100 drugs associated with more than 50 genes [1]. Of these drugs, 42 FDA cancer drugs include PGx information in their package inserts. Clearly, cancer therapy is one of the most intensively studied topics in PGx [2–4], and therefore relevant PGx data are accumulating quickly. Thus, it is critical to determine how to use and integrate cancer based PGx information effectively and to enable revelation of

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 639–651, 2014.
DOI: 10.1007/978-3-319-13186-3_57

hidden relationships among drug treatments, genomic variations, and phenotype traits to better support drug discovery and next generation of treatment. To our knowledge, no integration efforts have been directed specifically toward cancer based PGx. Suggested Ontology for Pharmacogenomics (SO-Pharm) [5] and Pharmacogenomics Ontology (PO) [6] are two existing ontologies for general PGx integration. They provided a first step toward integrating and representing PGx (and related) knowledge in the web ontology language (OWL), a web standard [7]. SO-Pharm contains so many classes and relations to represent generic PGx information that it is computationally expensive "and leads to significantly higher complexity for knowledge composition" [8]. It therefore presents challenges to users "in asserting knowledge or making routine queries" [8]. PO is a case-driven PGx data integration platform that aims to question-answering. Our study aims to integrate PGx information by focusing on oncology domain from diverse PGx resources. In addition, we will not only integrate existing PGx information, but also add inferred associations, which will support the novel indication detection for used drugs.

Idiosyncratic information without semantic interoperability and standard-based annotation, however, adds no value to the scientific commons. These idiosyncratic data must be annotated using standard terms and elements that correspond to the way scientists might search, integrate, inference, or expand upon the data. In the oncology community, the FDA and National Cancer Institute (NCI) attempt to document approved cancer drug information in a meaningful way. For instance, cancer drugs can be browsed by approved date with detailed description from the FDA; [9] they also can be queried/browsed by specific cancer type from the NCI [10], in which cancer drugs have been mapped to the NCI Thesaurus [2]. Nevertheless, to our knowledge, there is no data normalization effort made for cancer based PGx information. Lack of such effort hinders data sharing and further data integration. The CPN constructed in this study has been highlighted with normalization tags by leveraging the controlled terminologies and vocabularies.

In this study, we integrated multiple well known PGx resources including the PharmGKB [1], the FDA Pharmacogenomic Biomarkers in Drug Labeling [11], and the Catalog of Published Genome-Wide Association Studies [12] to construct a cancer based PGx network, named CPN. A majority of terms contained in this network have been represented with relevant standards. To demonstrate the capability of the CPN for drug repurposing, two case studies have been performed.

2 Materials

2.1 NCI Cancer List

National Cancer Institute (NCI) has maintained the alphabet links for information on a particular type of cancer. In this study, we have manually collected 160 distinct cancer types through de-duplication including bladder cancer, breast cancer, leukemia, and so on from NCI by Nov 14, 2013 [13].

2.2 Pharmacogenomics Knowledge Base (PharmGKB)

PharmGKB contains genomic, phenotype and clinical information collected from PGx studies. It provides information regarding variant annotations, drug-centered pathway,

Table 1. Examples of PGx associations extracted from the PharmGKB

Entity1_id	Entity1_ name	Entity1_ type	Entity2_id	Entity2_ name	Entity2_ type	PMIDs
PA443512	Urinary Bladder Neoplasms	Disease	rs762551	rs762551	VariantLocation	18798002
rs762551	rs762551	VariantLocation	PA443434	Arthritis, Rheumatoid	Disease	18496682
PA443434	Arthritis, Rheumatoid	Disease	PA27093	CYP1A2	Gene	18496682;19581389
PA27093	CYP1A2	Gene	PA450688	olanzapine	Drug	19636338;21519338

pharmacogenomic summaries, clinical annotations, PGx-based drug-dosing guidelines, and drug labels with PGx information [1]. In this study, we used PGx information extracted from a relationship file received from the PharmGKB by May 8, 2013, which provides associations between two PGx concepts, including drug, gene, disease, SNP and haplotype. Some examples are shown in Table 1. All fields listed in Table 1 were extracted and applied in this study.

The detailed information about individual disease, drug and gene terms were extracted from the corresponding *Disease*, *Drug* and *Gene* files downloaded from the PharmGKB by November 15, 2013 [14].

2.3 FDA Pharmacogenomic Biomarkers in Drug Labeling

The US Food and Drug Administration (FDA) provides a table of biomarkers for some FDA-approved drugs. The table contains “Therapeutic areas” field indicating the treatment intention of the drugs, such as “Oncology”, “Psychiatry”, and etc., as well as the “HUGO Symbol” field representing associated genes. In this study we extracted these two fields that are “Oncology” related. The table was downloaded by Dec 3, 2013 [9].

2.4 Catalog of Published Genome-Wide Association Studies

NIH provides a Catalog of Published Genome-Wide Association Studies (GWAS), which has identified single nucleotide polymorphisms (SNPs) and reported genes for major disease traits. We extracted cancers and related genes and SNPs from the “Disease/Trait”, the “Reported Gene(s)” and “SNPs” fields respectively. The Catalog was downloaded by Dec 3, 2013 [12].

2.5 National Center for Biomedical Ontology (NCBO)

The NCBO provides an ontology-based web service that can annotate public datasets with biomedical ontology concepts [15]. We used the NCBO Bioportal REST service [16] to access biomedical ontologies. In this study, we utilized this service to normalize drug and disease terms with Systematized Nomenclature of Medicine-Clinical Terms (SNOMED-CT) [17] and RxNorm [18].

3 Methods

To construct the CPN, we designed an approach that contains three steps: cancer based PGx association identification, cancer based PGx concept normalization, and the CPN construction. In the first step, we identified cancer based PGx associations from the PharmGKB, the GWAS Catalog and the FDA Biomarker table. Then we mapped cancer based PGx concepts to standard vocabularies, for instance, drugs to RxNorm, diseases to SNOMED-CT, genes to HUGO gene symbol and so on. Once the PGx associations were normalized, we built the CPN. Figure 1 presents the architecture of this approach. More details about each step and case studies will be described in the following sections.

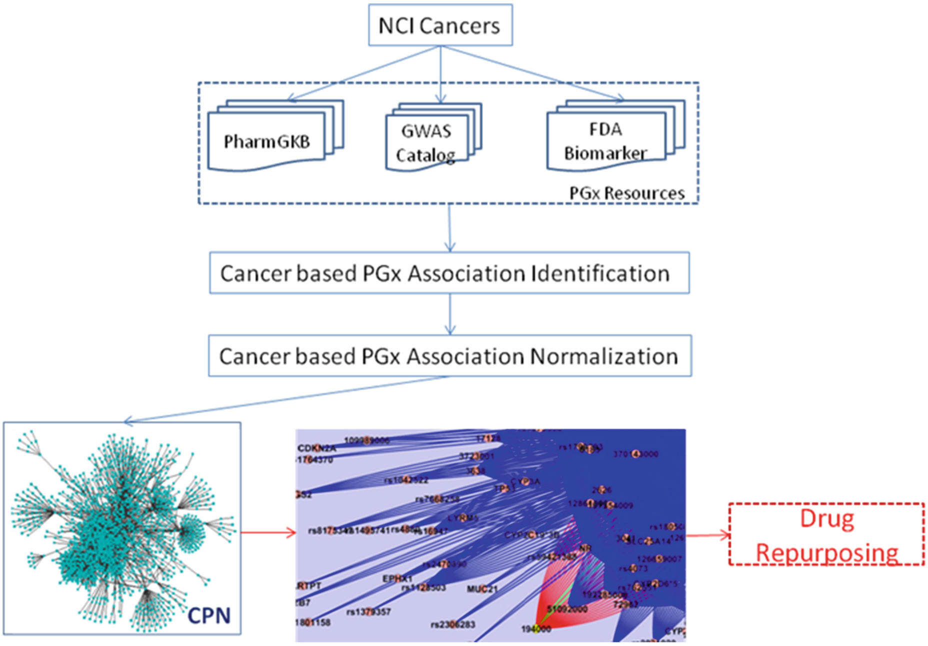

NCI Cancers
PharmGKB
GWAS Catalog
FDA Biomarker
PGx Resources
Cancer based PGx Association Identification
Cancer based PGx Association Normalization
CPN
Drug Repurposing

Fig. 1. The architecture of the approach being used for the CPN construction

3.1 Cancer Based PGx Association Identification

To extract cancer based PGx associations, we first manually searched for the 160 distinct NCI cancer terms that were manually collected through de-duplication and called as seeds against the PharmGKB. Once the seeds have been found in the PharmGKB dataset, we performed an iterative search to identify related PGx associations for these seeds. This search was not terminated until it accomplished the extraction of the fourth-degree concepts that are four nodes away from the seeds. In detail, starting from the seeds, we searched for the first-degree concepts that are directly connected to the seeds, the second-degree concepts that are the neighbors of the first degree concepts were retrieved afterwards, then the third-degree concepts that are the neighbors of the second-degree concepts, followed by the fourth–degree concepts. We iteratively extracted the associations related to these seeds from the fields listed in Table 1. For instance, beginning with the seed “Urinary Bladder Neoplasms”, we can iteratively find its associations, including SNP “rs762551” - “Urinary Bladder Neoplasms”, disease “Arthritis, Rheumatoid” - “rs762551”, gene “CYP1A2” - “Arthritis, Rheumatoid”, and drug “Olanzapine” - “CYP1A2”, which is shown in Table 1. These association pairs were the building blocks for constructing the CPN. Besides drug, disease and gene, we also extracted haplotype and SNP information that exist in the PharmGKB relationship file. To reflect an assumption that concepts with shorter distance to the seeds might have stronger associations with these seeds, we assigned different weight scores to the PGx concepts based on their degrees. The first degree concept was conferred with a higher weight score of “4”, then the second degree with “3”, the third degree with “2” and the fourth degree with “1”.

Additional PGx information available from the GWAS Catalog and the FDA biomarker table has also been extracted and imported into the CPN. We manually identified the seeds in the GWAS Catalog based on the NCI cancer terms. We then extracted the PGx associations related to the seeds from the fields of “Disease/Trait”, “Reported Gene(s)” and “SNPs” in the GWAS Catalog. Note that in this part we were not performing an iterative search to find other indirect associations, as we were interested in identifying and assigning a higher weight score for PGx associations co-occurring in the PharmGKB and this catalog. Meanwhile, we extracted PGx pairs between “Oncology” drugs and associated genes from the FDA biomarker table.

3.2 Cancer Based PGx Association Normalization

We normalized disease terms by SNOMED-CT [17], drugs by RxNorm [18], genes by the Human Genome Organization (HUGO) [19] gene symbols, SNP by the National Center for Biotechnology Information [20] reference SNP ID number (rsID). Genes, SNPs, haplotypes derived from the three resources have already been represented in standard forms. Therefore, no additional normalization has been performed accordingly. In this study, we primarily focused on the normalization for drug and disease terms.

Disease Term Normalization. The PharmGKB has provided manual annotations for disease terms with normalized vocabularies, including SNOMED-CT [17], Medical Subject Headings (MeSH) [21], Unified Medical Language System (UMLS) [22], etc., which are available in the downloadable Disease file. However, the mappings to SNOMED-CT are incomplete. Therefore, we normalized disease terms without SNOMED-CT codes by employing the NCBO Bioportal REST service [16] programmatically. The REST service accepts a list of disease names as input and outputs an XML file annotated with SNOMED-CT codes. A Java program has been written to automatically invoke this REST service and parse the XML file to retrieve SNOMED-CT codes. We also applied the NCBO REST service to map cancer terms from the GWAS catalog to SNOMED-CT. It is worthy to note that we specified “isexact-match = 1” as one of the input parameters when executing the NCBO REST service.

In another word, the mapped SNOMED-CT terms are exactly matched to the input disease names, thus, no additional evaluation is needed to validate the mapping performance. We manually checked and mapped the unmapped disease terms to SNOMED-CT.

Drug Term Normalization. The same mapping strategy has been applied to the diseases for drug terms, (1) we reused the normalized terms from the PharmGKB; (2) the NCBO Bioportal REST service was invoked to retrieve RxNorm Concept Unique Identifiers (RxCUIs) for those PharmGKB drugs and the drugs from the FDA biomarker table (no drug information in the GWAS catalog) that are without RxCUIs; (3) manual annotation was performed for unmapped drugs.

3.3 Cancer Based PGx Network Construction

Once the normalized cancer based PGx associations have been identified, we linked these associations with common concepts to construct the CPN. In the CPN, the nodes correspond to the individual cancer based PGx concepts including drug, gene, disease, SNP and haplotype. The edges correspond to the PGx associations. Table 2 shows the types of PGx associations contained in the CPN.

Enriched with PGx information from the PharmGKB, the GWAS catalog and the FDA biomarker table, the CPN could be used to infer novel PGx associations. We have explored Cytoscape [23] to visualize the CPN.

Table 2. Types of Association available in the CPN

Pairs Resources	Drug-Gene	Drug-Haplotype	Drug-Disease	Drug-SNP	Drug-Drug	Disease-SNP	Disease-Hyplotype	Gene-Disease	Gene-Gene	Gene-SNP
PharmGKB	√	√		√	√	√	√	√	√	
GWAS Catalog						√		√		√
FDA Biomarkers	√		√							

4 Results

4.1 Cancer Based PGx Association Identification

PharmGKB. Total 38 distinct seeds have been identified from the PharmGKB. Accordingly, we have extracted 2,964 concepts that are associated with these seeds, corresponding to 13,221 PGx pairs. Among these pairs, there are 402 drugs, 205 diseases, 825 genes, 1333 SNPs and 199 haplotypes.

Table 3 shows results of PGx associations extracted from the PharmGKB. For example, there are 38 seeds (cancer terms) associated with 393 Disease-Gene pairs, 37 Disease-Haplotype pairs and 530 Disease-SNP pairs. The numbers shown in Table 3 are unique.

Table 3. Results of PGx association extraction from the PharmGKB

Degree of concepts	Number of concepts	No. of pairs							
		Disease-Gene	Disease-Haplotype	Disease-SNP	Drug-Gene	Drug-Haplotype	Drug-SNPs	Drug-Drug	Gene-Gene
Seeds	38	393	37	530	0	0	0	0	0
1	605	1018	50	1155	1827	77	1607	0	195
2	735	1700	278	2483	2972	974	3716	1	944
3	2646	1705	277	2492	2965	974	3710	1	982
4	1196	0	0	0	0	0	0	0	0
Total	2964	1723	277	2500	3012	974	3718	1	1016

FDA Biomarkers and GWAS Catalog. We manually identified 42 oncology drugs from the FDA biomarker table. As some of drugs are associated with multiple genes, total 55 drug and gene pairs corresponding to 44 genes were extracted.

We extracted 31 cancers with the GWAS catalog, of which there are 2455 PGx pairs corresponding to 720 genes, 31 diseases and 598 SNPs.

4.2 Cancer Based PGx Association Normalization

Among 402 drugs extracted from the PharmGKB in this study, 323 were mapped to RxNorm by the PharmGKB. For the rest of 79 drugs without RxCUIs, 53 were mapped to RxNorm by invoking the NCBO REST service automatically. For 205 PharmGKB disease terms being used in this study, 186 have been mapped to SNOMED-CT by the PharmGKB, and another 10 were mapped to SNOMED-CT by calling the NCBO REST service programmatically. Out of 42 drugs from the FDA biomarker table, 41 were mapped to RxNorm by using NCBO REST service. Out of 31 cancer terms identified from the GWAS Catalog, 29 were mapped to SNOMED-CT by the NCBO REST service. At last, we manually reviewed 27 drugs and 11 disease unmapped to the standards, additional 5 drugs and 8 diseases were successfully mapped. Reasons for the failed mapping will be discussed in the discussion section.

In summary, 394 out of 416 (94.7 %) unique drug concepts have been mapped to RxNorm, and 215 out of 218 (98.6 %) unique disease concepts been mapped to SNOMED-CT.

4.3 Cancer Based PGx Network (CPN)

The CPN contains 4,342 distinct nodes and 15,600 pairs in total. A sub-network extracted from the CPN specifically for “urinary bladder cancer” is shown at the left lower corner of Fig. 1. The entire CPN can be visualized as cancer terms centralized concentric circles gradually expanded with cancer associated concepts.

4.4 Case Studies

The CPN provides comprehensive PGx information to support advanced cancer relevant research. Specifically, we can identify possible drug repurposing candidates from the CPN by utilizing network analysis approaches. The below two case studies illustrate how to leverage the information extracted from the CPN for drug repurposing.

The first case. "Paclitaxel" is used to treat Kaposi's sarcoma, as well as the lung, ovarian, and breast cancer, which is documented in the "Indications & Usage" section of the structured product label [24]. In this case study, we were interested in revealing the new indications of Paclitaxel from the CPN. We searched against the CPN with RxCUI "56946" (Paclitaxel), and obtained 78 directly related concepts, including "MTHFR" and "**rs1801133**", from which "Alzheimer Disease" with SNOMED-CT code, "26929004", have been identified. Figure 2 shows a zoomed out sub-network of Paclitaxel, where blue solid lines indicate the direct association existed in the CPN, while the red dotted line indicate the indirect inference applied in this case study.

As an SNP, rs1801133 is encoding a variant in the MTHFR gene, which encodes an enzyme involved in folate metabolism [24]. Then associations of Paclitaxel-MTHFR-"Alzheimer Disease", can be further validated by literatures as follows, (1) Paclitaxel potentiated the inhibitory effect of specific antisense against MTHFR on human colon and lung carcinoma xenografts [2, 25] (2) The severity and biochemical risk factors of Alzheimer's disease may be influenced by the MTHFR 677T allele in an Egyptian population [26] and the association between MTHFR A1298C polymorphisms as a possible risk factor and Alzheimer's disease was verified [27].

By conducting network analysis upon the CPN, we found that Paclitaxel is related to "Alzheimer Disease" through MTHFR and "**rs1801133**". Evidences are mounting in the literature that Alzheimer disease may be a new indication of the cancer drug Paclitaxel, for example Paclitaxel may rescue neurons from undergoing hallmark tau-induced Alzheimer disease cell pathologies [28] and Paclitaxel has the potential to treat Alzheimer disease [29]. That is to say,"Paclitaxel" may be a potential drug repurposing candidate for the treatment of Alzheimer Disease.

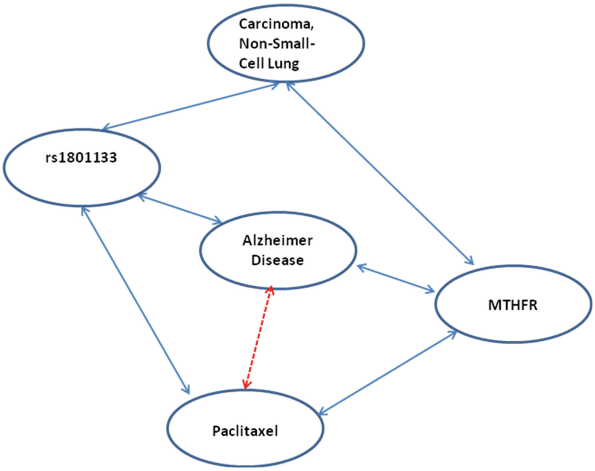


Fig. 2. A sub-network of Paclitaxel taken from the CPN

The second case. "Capecitabine" is originally indicated for the treatment of breast cancer and colorectal cancer as stated in the drug label [30]. In this case study, we were interested in alternative indication identification for Capecitabine. We first searched for the RxCUI, 194000 (Capecitabine) and extracted 51 related nodes including the gene "CYP1A1", from which related nodes including "Urinary Bladder Neoplasms" have been identified. A sub-network of Capecitabine visualized by Cytoscape in the CPN is shown at the right lower corner in Fig. 1, where the edges in red indicate all associations for Capecitabine (194000), and the green edges indicate associates from Capecitabine to DPYD and C18orf56.

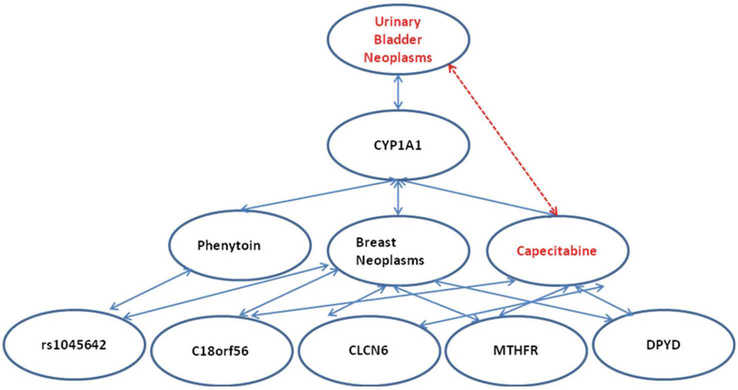


Fig. 3. A sub-network of Capecitabine taken from the CPN

Figure 3 shows a zoomed out sub-network of Capecitabine, where blue solid lines and the red dotted line represent the same meaning as in Fig. 2. The association between "Urinary Bladder Neoplasms" and "Capecitabine" could be inferred through multiple paths as shown in Fig. 3. Among all paths between these two, the shortest path is Capecitabine-CYP1A1-Urinary Bladder Neoplasms, of which the association could be proved from literatures: (1) "CYP1A1 rs1048943 A > G (Ile462Val) polymorphism is a potential prognostic marker for survival outcome after docetaxel plus capecitabine chemotherapy" [31]; (2) active CYP1A1 and CYP1B1 overexpression is revealed in bladder cancer [32]. (3) the combination of Capecitabine and radiation therapy offers a promising treatment option for bladder cancer patients who are not candidates for surgery or cisplatin-based chemotherapy [33]; (4) a patient with metastatic bladder cancer responded well to second-line capecitabine with a clinically meaningful progression-free survival [34]. Through this validation chain, the inference that the breast cancer drug, "Capecitabine" might be used for urinary bladder cancer could be made.

In conclusion, we identified "Capecitabine" as a potential drug repurposing candidate for the treatment of urinary bladder cancer from the CPN with strong evidence supports including multiple inference paths and confirming literatures.

5 Discussion

We have constructed the CPN by integrating the cancer based PGx information derived from three public PGx resources, the PharmGKB, the FDA biomarker table and the GWAS Catalog. The CPN built in this study offers comprehensive cancer based PGx information to support cancer orientated research, especially for drug repurposing.

5.1 Benefits Gained from the CPN

Supporting further data Integration. Data integration is essential in the big data era. It is important to aggregate different pieces of data from different areas to solve fundamental scientific questions. Particularly, in this study we have integrated data from various PGx data resources and built a cancer based PGx data repository. The concepts (nodes) included in the CPN were normalized with multiple standard biomedical terminologies and domain standards. The majority portion (99.4 %) of the concepts has been normalized, only 0.6 % of concepts failed to be normalized. Manual review for the failed mapping concepts showed that some drugs were unmapped due to chemical IUPAC names being used as drug names by the PharmGKB, which were not included in RxNorm, for example, “1-methyloxy-4-sulfone-benzene”. Or failures were resulted from drug class names being used, such as “Analgesics and Anesthetics” and “Anti-inflammatory and Antirheumatic Products”. In terms of diseases, the names of the unmapped diseases were either presented too-broadly, such as, “Substance-Related Disorders” or too narrowly, such as “Therapy-Related Acute Myeloid Leukemia”, that cannot be mapped to SNOMED-CT. Once the normalization task is accomplished, more relevant data can be deposited and integrated into the CPN, such as Electronic Medical Records (EHRs), DrugBank [35] and KEGG [36].

Supporting Oncology Based Drug Discovery. PGx data including the detailed information for drugs, diseases, genes, SNPs, etc., has been regarded as a basis for individualized medicine. While generic PGx data could be obtained publicly, drug, disease, gene, SNP and haplotype resources have not, as yet, been well-integrated to support the oncology based drug discovery. With various association types including Disease-Gene, Drug-Gene, etc. as shown in Table 2, the CPN can serve as a highly relevant cancer knowledge base and a valuable platform for oncology based research on drug repurposing. Thus, it would result in the shortening of the entire process for drug development, as our second case study has successfully proved such capability of the CPN. Additionally two advantages inherent in the CPN will strengthen its application in drug repurposing, including: (1) the CPN contains both direct and indirect cancer based PGx associations, thus, more drug candidates can be identified via automated inference; (2) a majority of concepts contained in the CPN are normalized with standard vocabularies, which enables further integration with other relevant resources to support more novel indication identifications.

5.2 Limitation and Future Study

Path Ranking. The current version of the CPN includes cancer based PGx information extracted from three major PGx resources. Although only 38 cancer terms have been found in the PharmGKB, 42 cancer drugs identified from the FDA biomarker table, and 31 cancer terms found from the GWAS catalog, the total number of nodes and edges of the CPN is 19,942, as we included all associations up to four nodes away from the cancer seeds. In this study, we focused on the CPN construction and the demonstration of the capability of the CPN. We did not work on path ranking to output a ranked list of paths that are associated with specific concepts from the CPN. However, when we conducted case studies, in order to filter out the most significant paths based on the queries, some initial ranking rules have been applied. For example, weight scores according to the degrees of concepts, path length, and VIP pairs from the PharmGKB have been applied for path ranking. In the future study, we will incorporate these rules with other ranking methods, such as PageRank [37], and genetic association p-values derived from GWAS [12], to output the most correlated paths for a particular query.

Disambiguating Drug-Disease Association. Detailed information on specifying drug and disease association is critical for drug repurposing, as we have to determine whether this drug is used to treat this disease or this drug may cause such a disease as an adverse drug event. Consequently, the novel indication may be identified for this drug for further evaluation. In this study, all drug and disease associations were directly extracted from the original resources, no additional step has been applied to disambiguate such associations. In our previous study, we have employed NDF-RT and SPLs to annotate drug and disease relationships in the PharmGKB [38]. We will apply the annotation results [38] along with the existing annotations from NDF-RT, ADEpedia [39], LinkedSPLs [40] into the future study, inserting a particular tag for differentiating indications and adverse drug events.

In this study we have integrated three existing PGx resources into the CPN, and successfully demonstrated its capability for drug repurposing. As more PGx resources are available publicly, such as DrugBank, KEGG, etc., we propose to extract further cancer based PGx information from them. Also we will identify PGx associations from pathways, and apply Natural Language Processing (NLP) [41] tools and algorithms to extract such associations from literatures. The ultimate goal will be leveraging semantic web technologies (SWT) [42] to present such comprehensive cancer based PGx information in RDF [43] or OWL [7], which can support automated inference for drug repurposing.

Acknowledgement. This work was supported by the Pharmacogenomic Research Network (NIH/NIGMS-U19 GM61388).

References

1. Hewett, M., Oliver, D.E., Rubin, D.L., et al.: PharmGKB: The pharmacogenetics knowledge base. Nucleic Acids Res. **30**(1), 163–165 (2002)

2. Fu, B., Brennan, R., O'Sullivan, D.: A configurable translation-based cross-lingual ontology mapping system to adjust mapping outcomes. Web Semant. Sci. Serv. Agents World Wide Web **15**, 15–36 (2012)
3. Lipson, D., Capelletti, M., Yelensky, R., et al.: Identification of new ALK and RET gene fusions from colorectal and lung cancer biopsies. Nat. Med. **18**(3), 382–384 (2012)
4. Kreso, A., O'Brien, C.A., van Galen, P., et al.: Variable clonal repopulation dynamics influence chemotherapy response in colorectal cancer. Science **339**(6119), 543–548 (2013)
5. Coulet, A., Smail-Tabbone M., Napoli, A., Devignes, M.D.: Suggested ontology for pharmacogenomics (SO-Pharm): Modular construction and preliminary testing. In: Proceedings of the Wokshop on Knowledge Systems in Bioinformatics, 29 October 2006
6. Dumontier, M., Villanueva-Rosales, N.: Towards pharmacogenomics knowledge discovery with the semantic web. Briefings Bioinform. **10**(2), 153–163 (2009)
7. McGuinness, D.L., Van Harmelen, F.: OWL web ontology language overview. W3C recommendation, vol. 10, pp. 2004–03 (2004)
8. Coulet, A., Smaïl-Tabbone, M., Napoli, A., Devignes, M.-D.: Suggested ontology for pharmacogenomics (SO-Pharm): Modular construction and preliminary testing. Paper presented at: On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops2006
9. FDA Table of Pharmacogenomic Biomarkers in Drug Labeling. http://www.fda.gov/drugs/scienceresearch/researchareas/pharmacogenetics/ucm083378.htm. Accessed 14 Jan 2014
10. Kumar, S.K., Harding, J.: Ontology mapping using description logic and bridging axioms. Comput. Ind. **64**, 19–28 (2012)
11. Frueh, F.W., Amur, S., Mummaneni, P., et al.: Pharmacogenomic biomarker information in drug labels approved by the United States food and drug administration: prevalence of related drug use. Pharmacother. J. Hum. Pharmacol. Drug Therapy **28**(8), 992–998 (2008)
12. Hindorff, L.A., MJEBI, Morales, J., Junkins, H.A., Hall, P.N., Klemm, A.K., Manolio, T.A.: A Catalog of published genome-wide association studies. www.genome.gov/gwastudies. Accessed 14 Jan 2014
13. National Cancer Institute. http://www.cancer.gov/cancertopics/types/alphalist. Accessed 14 Jan 2014
14. The Pharmcogenomics Knowledgebase. http://www.pharmgkb.org/. Accessed 14 Jan 2014
15. Whetzel, P.L., Noy, N.F., Shah, N.H., et al.: BioPortal: Enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. Nucleic Acids Res. **39**(suppl 2), W541–W545 (2011)
16. National Center for Biomedical Ontology Rest Service. http://data.bioontology.org/documentation. Accessed 14 Jan 2014
17. Bos, L.: SNOMED-CT: The advanced terminology and coding system for eHealth. Med. Care Compunetics 3 **121**, 279 (2006)
18. Nelson, S.J., Zeng, K., Kilbourne, J., Powell, T., Moore, R.: Normalized names for clinical drugs: RxNorm at 6 years. J. Am. Med. Inform. Assoc. **18**(4), 441–448 (2011)
19. Gardiner, K.: Human genome organization. Curr. Opin. Genet. Dev. **5**(3), 315–322 (1995)
20. Sayers, E.W., Barrett, T., Benson, D.A., et al.: Database resources of the national center for biotechnology information. Nucleic Acids Res. **39**(suppl 1), D38–D51 (2011)
21. Coletti, M.H., Bleich, H.L.: Medical subject headings used to search the biomedical literature. J. Am. Med. Inform. Assoc. **8**(4), 317–323 (2001)
22. Bodenreider, O.: The unified medical language system (UMLS): Integrating biomedical terminology. Nucleic Acids Res. **32**(suppl 1), D267–D270 (2004)
23. Saito, R., Smoot, M.E., Ono, K., et al.: A travel guide to Cytoscape plugins. Nat. Methods **9** (11), 1069–1076 (2012)
24. Drugs@FDA. http://www.accessdata.fda.gov/scripts/cder/drugsatfda/

25. Stankova, J., Shang, J., Rozen, R.: Antisense inhibition of methylenetetrahydrofolate reductase reduces cancer cell survival in vitro and tumor growth in vivo. Clin. Cancer Res. **11**(5), 2047–2052 (2005)
26. Elhawary, N.A., Hewedi, D., Arab, A., et al.: The MTHFR 677T allele may influence the severity and biochemical risk factors of Alzheimer's disease in an Egyptian population. Dis. Mark. **35**(5), 439–446 (2013)
27. Mansouri, L., Fekih-Mrissa, N., Klai, S., Mansour, M., Gritli, N., Mrissa, R.: Association of methylenetetrahydrofolate reductase polymorphisms with susceptibility to Alzheimer's disease. Clin. Neurol. Neurosurg. **115**(9), 1693–1696 (2013)
28. Shemesh, O.A., Spira, M.E.: Rescue of neurons from undergoing hallmark tau-induced Alzheimer's disease cell pathologies by the antimitotic drug paclitaxel. Neurobiol. Dis. **43** (1), 163–175 (2011)
29. Zhang, B., Maiti, A., Shively, S., et al.: Microtubule-binding drugs offset tau sequestration by stabilizing microtubules and reversing fast axonal transport deficits in a tauopathy model. Proc. Natl. Acad. Sci. U.S.A. **102**(1), 227–231 (2005)
30. Drug label of Capecitabine in DailyMed. http://dailymed.nlm.nih.gov/dailymed/lookup.cfm?setid=a1de8bba-3b1d-4c9d-ab8a-32d2c05e67c8. Accessed 14 Jan 2014
31. Dong, N., Yu, J., Wang, C., et al.: Pharmacogenetic assessment of clinical outcome in patients with metastatic breast cancer treated with docetaxel plus capecitabine. J. Cancer Res. Clin. Oncol. **138**(7), 1197–1203 (2012)
32. Androutsopoulos, V.P., Spyrou, I., Ploumidis, A., et al.: Expression Profile of CYP1A1 and CYP1B1 Enzymes in Colon and Bladder Tumors. PLoS ONE **8**(12), e82487 (2013)
33. Patel, B., Forman, J., Fontana, J., Frazier, A., Pontes, E., Vaishampayan, U.: A single institution experience with concurrent capecitabine and radiation therapy in weak and/or elderly patients with urothelial cancer. Int. J. Radiat. Oncol. Biol. Phys. **62**(5), 1332–1338 (2005)
34. Michels, J., Barbour, S., Cavers, D., Chi, K.N.: Metastatic signet-ring cell cancer of the bladder responding to chemotherapy with capecitabine: Case report and review of literature. Can. Urol. Assoc. J. **4**(2), E55 (2010)
35. Wishart, D.S., Knox, C., Guo, A.C., et al.: DrugBank: A comprehensive resource for in silico drug discovery and exploration. Nucleic Acids Res. **34**(suppl 1), D668–D672 (2006)
36. Kanehisa, M., Goto, S.: KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res. **28**(1), 27–30 (2000)
37. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the web (1999)
38. Zhu, Q., Freimuth, R.R., Pathak, J., Durski, M.J., Chute, C.G.: Disambiguation of PharmGKB drug-disease relations with NDF-RT and SPL. J. Biomed. Inform. **46**(4), 690–696 (2013)
39. Jiang, G., Solbrig, H.R., Chute, C.G.: ADEpedia: a scalable and standardized knowledge base of Adverse Drug Events using semantic web technology. Paper presented at: AMIA Annual Symposium Proceedings (2011)
40. Hassanzadeh, O., Zhu Q, Freimuth, R., Boyce, R.: Extending the "Web of Drug Identity" with Knowledge Extracted from United States Product Labels. Submitted to AMIA Summit on Clinical Research Informatics (2013)
41. Chowdhury, G.G.: Natural language processing. Ann. Rev. Inf. Sci. Technol. **37**(1), 51–89 (2003)
42. Shenoy, M.K., Shet, K., Acharya, D.U.: Semantic Web Technologies (2010)
43. Klyne, G., Carroll, J.J., McBride, B.: Resource description framework (RDF): Concepts and abstract syntax, W3C recommendation, p. 10 (2004)

Linked Vaccine Adverse Event Data Representation from VAERS for Biomedical Informatics Research

Cui Tao[1](✉), Puqiang Wu[2], and Yuji Zhang[3,4]

[1] School of Biomedical Informatics, University of Texas Health Science Center at Houston, Houston, TX, USA
cui.tao@uth.tmc.edu
[2] University of Wisconsin-Madison, Madison, WI, USA
[3] Division of Biostatistics and Bioinformatics, University of Maryland Greenebaum Cancer Center, Baltimore, MD, USA
[4] Department of Epidemiology and Public Health, University of Maryland School of Medicine, Baltimore, MD, USA

Abstract. Vaccines have been one of the most successful public health interventions to date. The use of vaccination, however, also comes with possible adverse events. The U.S. FDA/CDC Vaccine Adverse Event Reporting System (VAERS) currently contains more 200,000 reports for post-vaccination events that occur after the administration of vaccines licensed in the United States. Although the data from VAERS has been applied to many public health and vaccine safety studies, each individual report does not necessary indicate a casuality relationship between the vaccine and the reported symptoms. Further statistical analysis and summarization needs to be done before this data can be leveraged. In this paper, we introduces our preliminary work on summarzing the VAERS data and representing the vaccine-symptom correlations as well as the meta data of their relations using RDF. We then apply network analysis approaches to the RDF data to illustrate a use case of the data. We further discuss our vision on integrating the data with vaccine information from other sources using RDF linked approach to faciliate more comprehensive analyses.

1 Introduction

Vaccines have been one of the most successful public health interventions to date with most vaccine-preventable diseases having declined in the United States by at least 95–99 %. The use of vaccination, however, also comes with possible adverse events. The U.S. FDA/CDC Vaccine Adverse Event Reporting System (VAERS) is a national vaccine safety surveillance program for post-vaccination adverse events (AE) that occur after the administration of vaccines licensed in the United States [1]. Currently the VAERS contains more than 200,000 reports in total. Patients or healthcare providers submit reports about cases of adverse events they have experienced on the VAERS website by providing information ranging from vaccine type, gender, age, symptoms and detailed description of occurred symptoms to onset dates, life-threatening status, hospitalization status, and death-status. The objectives of VAERS are to detect new, unusual, or rare vaccine adverse events; determine patient risk factors for

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 652–661, 2014.
DOI: 10.1007/978-3-319-13186-3_58

particular types of adverse events; identify vaccine lots with increased numbers or types of reported adverse events; and assess the safety of newly licensed vaccines [1].

Although a report was submitted into the VAERS system, that by no means is an absolute declaration that the vaccine had direct correlation with the reported symptoms. The causality relationship between a vaccine and an adverse event cannot be simply assumed by the VAERS report. In this study, we do not only focus on the raw data from the VAERS system, but also the correlation of vaccines and symptoms. Through statistical analysis, the correlation can be determined by relating frequency of a specific symptom to the corresponding vaccine and the related symptom with all the vaccines in the system. We represent information obtained and summarized from the VAERS database in the Resource Description Framework (RDF) format to facilitate further integration with other vaccine relevant data for more comprehensive analysis. Armed with such knowledge, the ability to predict adverse events, or to design new vaccine approaches that minimize or eliminate serious vaccine-related reactions could be devised, consistent with a more personalized or individual approach to vaccine practice.

After the vaccine-adverse event correlations are identified, how to organize these high-dimensional correlation data and facilitate pattern recognition by clinician experts is still a big challenge. In recent years, network analysis emerges as a very promising approach to address this. Network analysis allows simultaneous representation of complex associations (e.g., protein-protein interactions) among key elements (e.g., gene or proteins) in a system (e.g., gene regulatory networks). For example in the social networks, the nodes are individuals, organizations, or even the entire societies, and the edges are social relationships between the nodes. During last two decades, network-based computational approaches gained popularity and have become a new paradigm to investigate associations among biological entities (e.g., drugs, diseases, and genes). Applications of these approaches include drug repositioning [2, 3], disease gene prioritization [4–6], and identification of disease relationships [7, 8]. These network analysis approaches are usually developed based on the observations from real-world networks. First, most real-world networks (e.g., WWW network, protein-protein interaction network, and social network) are not randomly organized but are driven by preferential attachment and growth (e.g., some nodes have more connections than others). Such networks are called "Scale-free" networks. In the "scale-free" network, the most highly connected nodes are called "hub" nodes. Second, most real world networks are modular, comprised of small, densely connected groups of nodes. Network analysis metrics and algorithms have been designed to identify network hub nodes and modules in a scale-free network. For instance, in our previous work, we developed a network analysis approach to identify vaccine-related networks and their underlying structural information from PubMed literature abstracts, which were consistent with that captured by the Vaccine Ontology (VO) [9]. The modular structure and hub nodes of these vaccine networks reveal important unidentified knowledge critical to biomedical research and public health and to generate testable hypotheses for future experimental verification.

The rest of the paper is organized as follows. In Sect. 2, we discuss our methodology on data collection, summarization, representation, and analysis. In Sect. 3, we discuss the result of our preliminary study. In Sect. 4, we introduce our vision on further integrating the VAERS data with more vaccine data sources. Finally in Sect. 5, we conclude the paper and discuss future directions.

2 Methods

2.1 The VAERS Data Preparation

All of the VAERS data was downloaded from the reporting system's website (http://vaers.hhs.gov/index). All of the necessary files were downloaded from the reporting system in zip files from 1990 to 2013 and loaded into a MySQL relational database. More specifically, three tables are included in the database: Data, Vaccine, and Symptom. The Data table contains information including VAERS ID, date the report was received, the state patient was in, age of patient, sex, and detailed description of the symptom (e.g., if the symptom was life threatening, if the patient in the report died and if-so the date of death, if the patient ever attend the ER for treatment, and if so, how many days was the patient administered at the hospital.) The Vaccine table includes information about the vaccine administered to the patient such as vaccine manufacturer, type of vaccine, dosage of the vaccine, vaccination route, vaccination site, and vaccination name. Vaccine types are annotated with Vaccine Code. The Symptom table contains a list of symptom terms (MedDRA terms) involved in the report. Completed information about one report can be jointed from the three tables using VAERS ID.

2.2 The VAERS Data Summarization

As we discussed before, the VAERS is a spontaneous reporting system which contains unverified reports with inconsistent data quality. Symptoms reported occurring after vaccination do not necessarily have a causality association with the vaccine. In addition to the raw data downloaded from VAERS, we also used statistical methods to summarize meta-level features of vaccine-symptom pairs. For each vaccine-symptom pair, we calculated the following features (1) the number of reports that contains the pair; (2) the number of reports by year that contain the pair; (3) the demographic distribution among the reports that contain the pair (total and yearly) grouped by gender and age groups; and (4) overall proportional reporting ratio (PRR) and yearly PRRs [10]. A PRR is the ratio between the frequency with which a specific symptom (adverse event) occurs for a vaccine of interest (relative to all symptoms reported for the vaccine) and the frequency with which the same symptom occurs for all vaccines reported to the VAERS (relative to all symptoms for all vaccines reported to VAERS) [11]. A PRR greater than 1 suggests that the post-vaccination symptom (adverse event) is more commonly observed for individuals administrated with the particular vaccine, relative to all other vaccines reported to the VAERS.

2.3 RDF Representation

We represented vaccine-symptom pairs as well as the summarization features in Resource Description Framework (RDF). RDF is a W3C standard that specifies a graph-based data model for representing data. Each piece of information is represented as a triple: subject, predicate and object. The RDF representations will allow efficient querying and visualization of relationships between important biomedical entities. A distinguishing characteristic of RDF and ontologies compared to the conventional relational database is "their degree of connectedness, their ability to model coherent, linked relationships" [12]. After representing the associations using RDF graphs, it will enable us to leverage existing Semantic Web tools to explore the Semantic Web Linked Data in a flexible and scalable way. Moreover, it will enable powerful data integration among heterogeneous data sets, which is a well-known challenge in the translational science study community.

Figure 1 shows a sample RDF graph representation of vaccine adverse event associations. Using RDF, we can represent the association of a vaccine and an adverse event annotated with meta-information of the association. Each vaccine and AE will be assigned with a unique identifier. Using RDF reification, each association is also assigned with a unique identifier. The summarization data we collected from the previous section can be used to annotate the associations.

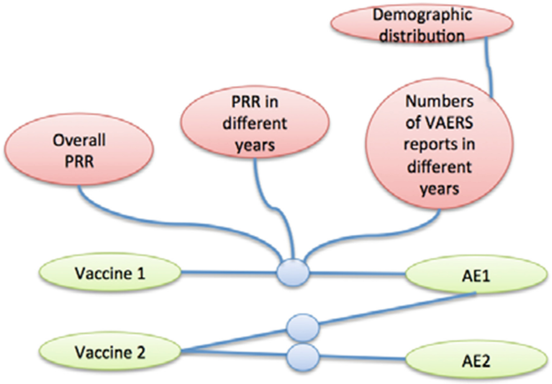


Fig. 1. Sample RDF graph representation of vaccine adverse event association

2.4 Network Analysis

The network analysis and visualization was performed in the Cytoscape tool [13]. Cytoscape is an open-source platform for integration, visualization, and analysis of biological networks. Its functionalities can be extended through Cytoscape plugins. Scientists from different research fields have contributed more than 160 useful plugins so far. These comprehensive features allow us to perform thorough network-level analyses, visualization of our association tables, and integration with other biological networks in the future. We used NetworkAnalyzer plugin (http://med.bioinf.mpi-inf.mpg.de/netanalyzer//index.php) to calculate average node degree, average path length, and network diameter for each vaccine-adverse event network generated from VAERS.

3 Result and Use Case Discussion

Overall, we have extracted 2,346,367 pairs of vaccine-symptom combinations from the VAERS system, with 83,148 distinct pairs. Over a 23-year period, 72 different vaccines and 5441 different adverse events were identified to have significant associations, i.e., associations reported in at least one year report in the system. The average shortest path and the network diameter were 2.58 and 6, respectively (Table 1). This demonstrates that vaccine-adverse event network is dense network, with any given node connected to all other nodes through an average of approximately two other nodes and a maximum of six nodes. This is explained partly that many vaccines are coadministered. However, given that there are more adverse events than vaccines in the network, it is plausible that many adverse events were reported together.

Table 1. General characteristics of the networks

	N_{node}	N_{link}	Average degree	Average path length	Network diameter
Overall	298	2786	13.87	2.58	6
1990	75	92	2.45	5.43	13
2000	98	116	2.37	4.81	10
2010	123	139	2.26	5.1	13

We further investigated how the vaccine-adverse event network evolves during three decades. Figures 2, 3, 4 represent the vaccine-adverse event networks extracted from VAERS reports submitted in 1990, 2000, and 2010. The overall network properties of three networks were shown in Table 1. Among three networks, 14 nodes were in all three networks, including 6 vaccines and 8 adverse events (Fig. 5). Haemophilus B Polysaccharide Vaccine (HBPV) is used for a routine immunization of children

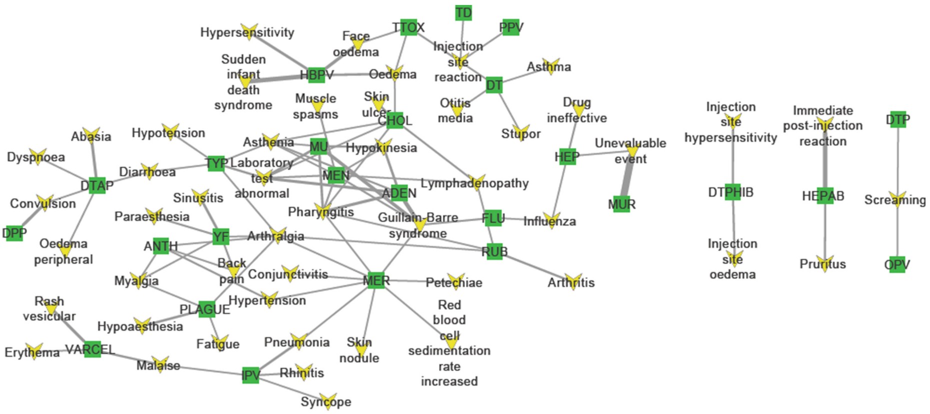


Fig. 2. Vaccine-adverse event network reconstructed from extracted from VAERS reports in 1990. Green rectangle: vaccine; yellow vee: adverse event. Edge: vaccine-adverse event association with PRR ratio greater than 1. Edge width denotes the PRR ratio of the edge (Color figure online).

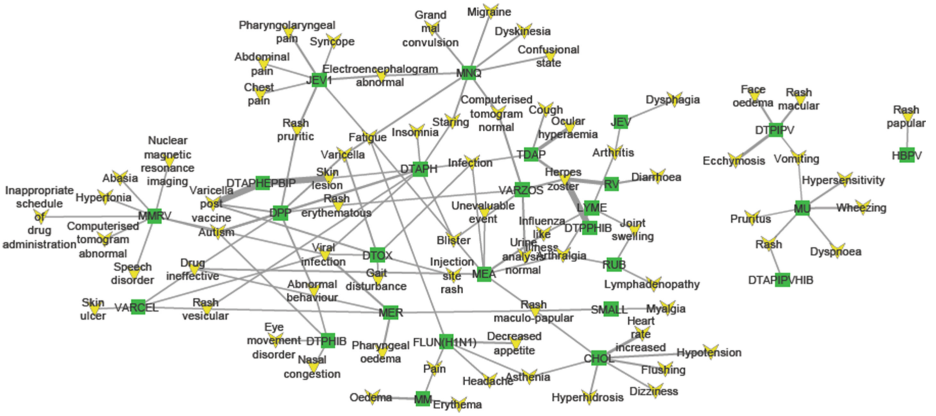


Fig. 3. Vaccine-adverse event network reconstructed from extracted from VAERS reports in 2000. Green rectangle: vaccine; yellow vee: adverse event. Edge: vaccine-adverse event association with PRR ratio greater than 1. Edge width denotes the PRR ratio of the edge (Color figure online).

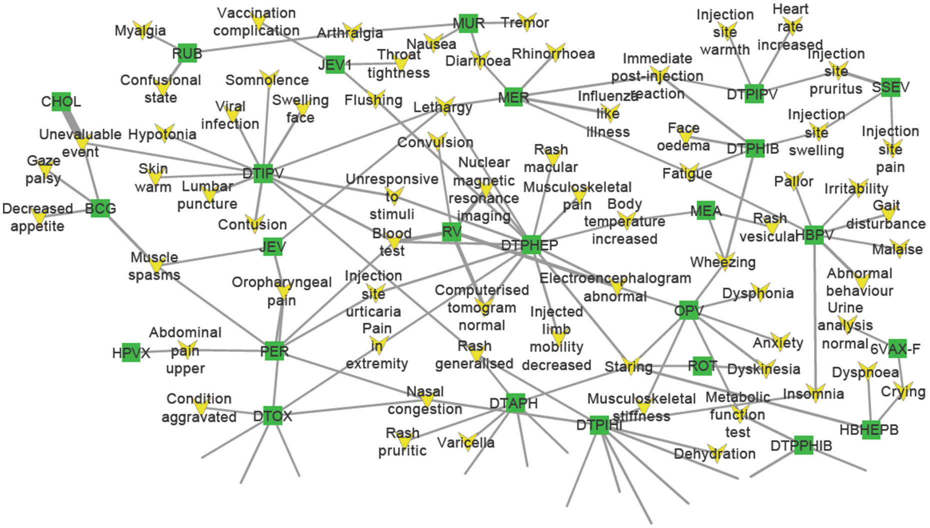


Fig. 4. Vaccine-adverse event network reconstructed from extracted from VAERS reports in 2010. Green rectangle: vaccine; yellow vee: adverse event. Edge: vaccine-adverse event association with PRR ratio greater than 1. Edge width denotes the PRR ratio of the edge (Color figure online).

24 months to 5 years of age. In three years we investigated, HBPV was associated with different adverse events with significant PPR ratios, suggesting that HBPV may cause different adverse events in different years. This could be due to the difference how this vaccine was manufactured in these years. Such network analysis can help clinician

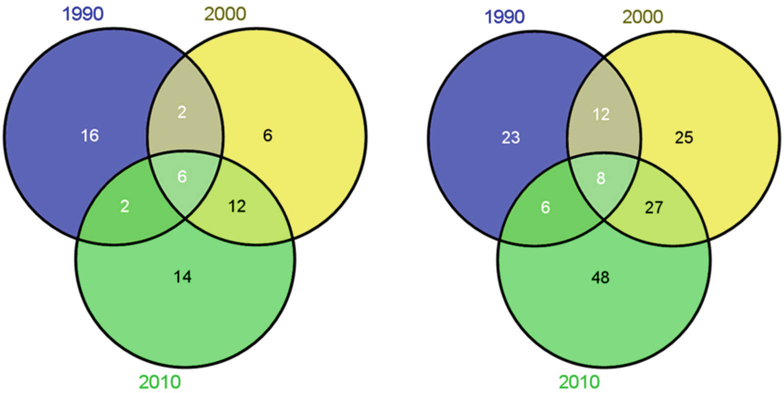


Fig. 5. Venn diagram of nodes among 1990, 2000, and 2010. (a) vaccine nodes; (b) adverse event nodes.

experts easily identify such differences and design experiments to further investigate the underlying biological mechanisms.

4 Linking with Other Resources

One unique benefit of RDF representation is that it provides a flexible way to link data from different sources together. With current technologic advances such as high throughput sequencing, transcriptomics, epigenetics, and proteomics, there are big amount of amount of data available for better understanding associations and mechanisms of VAEs and immunogenicity. With the RDF representation, we can intergrade the VAERS data with data from other sources such as PubMed literature, Vaccine Label data, and Vaccine ontology to create a Linked VAE data repository. Figure 6 shows the overview. For PubMed data, we have created the SemMed-RDF repository for representing associations among genetic factors, diseases, and drugs extracted from PubMED abstracts [14] based on the Semantic MEDLINE database [15]. This knowledgebase currently contains 843 k disease-disease, 111 k disease-gene, 1277 k disease-drug, 248 k drug-gene, 1900 k drug-drug, and 49 k gene-gene associations, annotated with their provenance information. We have the identified vaccine relevant associations with diseases, symptoms, and genes from SemMed-RDF [16]. This data can be integrated with the VAERS RDF data. In addition, we can also link vaccine relevant information from publicly available ontologies such as Vaccine Ontology (VO) [17] and the Ontology of Vaccine Adverse Event (OVAE) [18]. VO has modeled and classified various vaccines, including all licensed vaccines used in the USA. For each licensed vaccine, VO includes vaccine name, disease or pathogen name, manufacturer, CDC CVX (Codes for Vaccine Administered), host species (e.g., human), vaccine type based on preparation (e.g., killed or inactivated vaccine), vaccine antigen component, and vaccination route). The hierarchy structure of the vaccine in VO classifies the vaccine type based on pathogen taxonomy. OVAE is an ontology that represents and classifies the adverse events recorded in package insert (vaccine label) documents of commercial vaccines licensed by the USA Food and Drug Administration (FDA). Combined these sources, we can create the linked VAE Data, a centralized comprehensive knowledgebase for vaccines and their associations with genetic factors, diseases, and AE can be generated for large-scale computational studies of VAE mechanisms.

5 Conclusion and Future Work

In this paper, we discussed our preliminary effort on representing data summarized from VAERS database using RDF. We then applied network analysis on top of the data to illustrate how network-based analysis can be applied to identify underlying association patterns among vaccines and adverse events. Based on this preliminary work, we plan to extend this work in future research including: (1) identification of network modules in the vaccine-adverse event network; (2) investigation of vaccine-vaccine

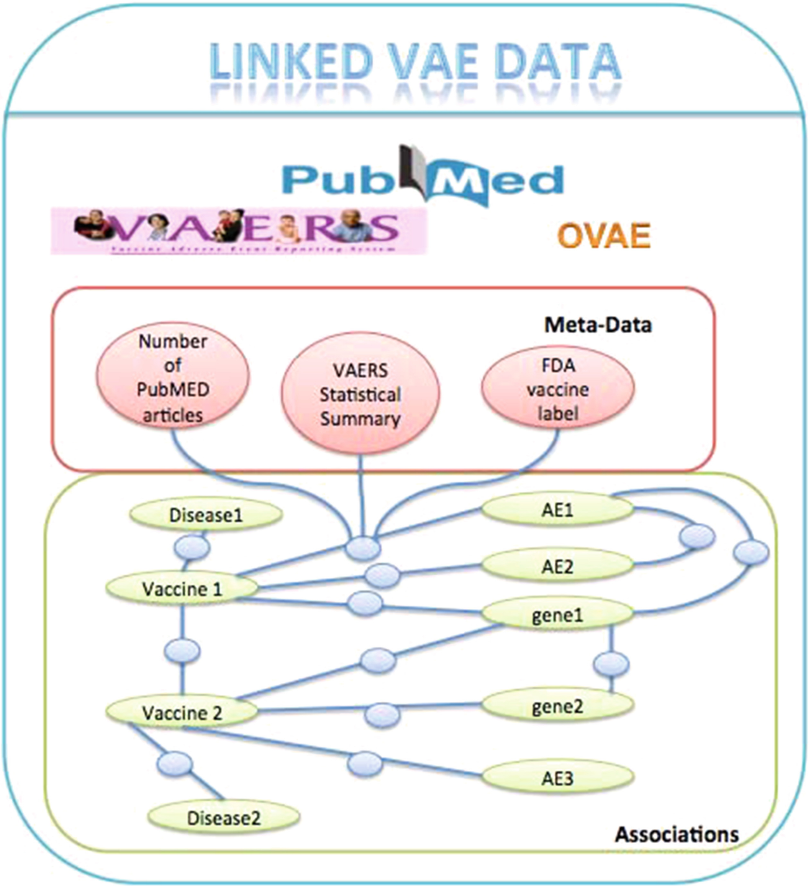


Fig. 6. Linked VAE data overview

associations by bipartite network projection strategy; (3) incorporation of more comprehensive vaccine-disease association databases (e.g., Semantic MEDLINE database) to construct more complete vaccine-related networks.

Acknowledgement. YZ's effort is supported by the Center for Individualized Medicine at Mayo Clinic.

References

1. The U.S. FDA/CDC Vaccine Adverse Event Reporting System (VAERS). http://vaers.hhs.gov/index
2. Arrell, D.K., Terzic, A.: Network systems biology for drug discovery. Clin. Pharmacol. Ther. **88**, 120–125 (2010)
3. Dudley, J.T., Deshpande, T., Butte, A.J.: Exploiting drug-disease relationships for computational drug repositioning. Briefings Bioinf. **12**, 303–311 (2011)
4. Piro, R.M., Di Cunto, F.: Computational approaches to disease-gene prediction: rationale, classification and successes. FEBS J. **279**, 678–696 (2012)
5. Kohler, S., Bauer, S., Horn, D., Robinson, P.N.: Walking the interactome for prioritization of candidate disease genes. Am. J. Hum. Genet. **82**, 949–958 (2008)
6. Chen, J., Aronow, B.J., Jegga, A.G.: Disease candidate gene identification and prioritization using protein interaction networks. BMC Bioinf. **10**, 73 (2009)
7. Goh, K.I., Cusick, M.E., Valle, D., Childs, B., Vidal, M., Barabasi, A.L.: The human disease network. Proc. Natl. Acad. Sci. U.S.A. **104**, 8685–8690 (2007)
8. Suthram, S., Dudley, J.T., Chiang, A.P., Chen, R., Hastie, T.J., Butte, A.J.: Network-based elucidation of human disease similarities reveals common functional modules enriched for pluripotent drug targets. PLoS Comput. Biol. **6**, e1000662 (2010)
9. Zhang, Y., Tao, C., He, Y., Kanjamala, P., Liu, H.: Network-based analysis of vaccine-related associations reveals consistent knowledge with the vaccine ontology. J. Biomed. Semant. **4**, 33 (2013)
10. Rothman, K.J., Lanes, S., Sacks, S.T.: The reporting odds ratio and its advantages over the proportional reporting ratio. Pharmacoepidemiol. Drug Saf. **13**, 519–523 (2004)
11. Banks, D., Woo, E.J., Burwen, D.R., Perucci, P., Braun, M.M., Ball, R.: Comparing data mining methods on the VAERS database. Pharmacoepidemiol. Drug Saf. **14**, 601–609 (2005)
12. An Executive Intro to Ontologies (2009). http://www.mkbergman.com/900/an-executive-intro-to-ontologies/
13. Smoot, M.E., Ono, K., Ruscheinski, J., Wang, P.L., Ideker, T.: Cytoscape 2.8: New features for data integration and network visualization. Bioinformatics **27**, 431–432 (2011)
14. Tao, C., Zhang, Y., Jiang, G., Bouamrane, M., Chute, C.: Optimizing semantic MEDLINE for translational science studies using semantic web technologies. In: International Conference on Information and Knowledge Management, Maui, HI (2012)
15. Rindflesch, T.C., Kilicoglu, H., Fiszman, M., Rosemblat, G., Shin, D.: Semantic MEDLINE: An advanced information management application for biomedicine. Inf. Serv. Use **31**, 15–21 (2011)
16. Zhang, Y., Tao, C., He, Y., Kanjamala, P., Liu, H.: Network-based analysis of vaccine-related associations reveals consistent knowledge with the vaccine ontology. J. Biomed. Semant. **4**, 33 (2013)

17. He, Y., Cowell, L., Diehl, A., Mobley, H., Peters, B., Ruttenberg, A., Scheuermann, R., Brinkman, R., Courtot, M., Mungall, C.: VO: Vaccine Ontology. In: The 1st International Conference on Biomedical Ontology(ICBO 2009) (2009)
18. Marcos, E., He, Y.: The Ontology of Vaccine Adverse Events (OVAE) and its usage in representing and analyzing vaccine adverse events. In: International Conference on Biomedical Ontology (2013) (accepted)

Pattern Mining and Application of Big Data

QRM: A Probabilistic Model for Search Engine Query Recommendation

JianGuo Wang[1,2](✉) and Joshua Zhexue Huang[1,3]

[1] Shenzhen Key Laboratory of High Performance Data Mining, Chinese Academy of Sciences, Shenzhen Institutes of Advanced Technology, Shenzhen 518055, China
wangjg@siat.ac.cn

[2] Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen 518055, China

[3] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
zx.huang@szu.edu.cn

Abstract. This paper proposes a query ranking model (QRM) for query recommendation to the Web users of a search engine. Given an initial query in a search session, a set of queries for the user to select as the next query are ranked based on the joint probability that the query is to be selected by the user and that the result of the query is to be clicked by the user, and that the clicked result will satisfy the user's information requirement. We define three utilities to solve the model, including a query level utility and two document level utilities that are the perceived utility representing user's action on the query result and the posterior utility representing user's satisfaction on the search result. We present the methods to compute the three utilities from the query log data. Experiment results on real query log data have demonstrated that the proposed query ranking model outperformed six baseline methods in generating recommendation queries.

Keywords: Query recommendation · Query log analysis · Query ranking · Query utility

1 Introduction

Search engines such as Google[1], Yahoo![2], and Bing[3] have become ubiquitous tools for people to find information from the Web. A search task starts with a query (usually one or a sequence of key words) issued to a search engine; the search engine processes the query and returns a web page showing an ordered list of sites which may contain the information the user wants; by viewing the brief information of the sites in the list, the user chooses a site to click, browses the documents at the open site to find the information in need or issues another

[1] http://www.google.com/

[2] http://www.yahoo.com/

[3] http://www.bing.com/

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 665–676, 2014.
DOI: 10.1007/978-3-319-13186-3_59

query to the search engine if the user is not satisfied. We call this process a query cycle, which represents the basic steps of using search engines.

In reality, a search task can take many query cycles and long time to complete. The query cycles for a search task form a search session. For a complicated search task, it is often difficult for the user to issue a right query to find the wanted information immediately. The user needs to go through a query refinement process, by trying different queries, to obtain the final result. To assist users to shorten search sessions and find the results as quickly as possible, query recommendation technologies have been developed in many search engines to improve Web search usability.

Existing query recommendation methods are based on similarity measures between queries. Given a query q, the candidate queries $\{q_1, q_2, \ldots, q_m\}$ for recommendation are ranked with respect to the similarity measure $S(q, q_i)$ where S is computed from the query log data. The top $k(< m)$ most similar queries are recommended to the user in the return page of q by the search engine. Different log data are used to compute the similarity between two queries, such as common clicked URLs [1] and consecutive reformulated queries [2]. The main problem of this similarity approach is that although similar queries $q_1, q_2, \ldots, q_k$ were recommended to q, however, whether the query's result would satisfy user's information need is unsure. Often, the results of several recommended queries are irrelevant and need to be browsed and then search sessions are prolonged.

A promising approach for query recommendation was recently proposed to model the expected information of query q and user's satisfaction on the information obtained from candidate queries [11]. This approach uses two utility concepts. Perceived utility is defined as the probability that the user will click a query result. Posterior utility is referred to as the useful information that the user can obtain from the clicked results. From these utility concepts, a dynamic Bayesian model QUM was built from search log data [11]. The experimental results have shown that the recommended queries could find more relevant documents.

The utility concepts in the QUM model were defined on the results of queries. They are referred as document-level utilities. The QUM model has ignored the query itself, i.e., the key word(s) which contains important information to query recommendation. QUM made two assumptions that all queries in a search session were issued from the same information need and the user was satisfied when the user clicked the result of the last query in the search session. These two assumptions do not accurately reflect the reality, which affects the performance of QUM.

In this paper, we propose a query ranking model QRM for query recommendation. The model is used to rank a set of queries which the user can select from the return page of an initial query as the next query to issue to the search engine. The ranking is based on the joint probability that a query is to be selected by the user and that the result of the query is to be clicked by the user, and that the user will be satisfied with the clicked result. We introduce three utilities to solve the model. Two document level utilities are the perceived utility representing user's action on the query result and the posterior utility representing user's satisfaction on the search result. A query-level utility represents attractiveness of a candidate query to the user. We propose three methods to compute the utilities from query log data and a new method to measure the satisfaction of a search session.

We used "Spring 2006 Data Asset" publicly available query log data from Microsoft Research to evaluate the QRM model. In the experiments, we chose six existing recommendation methods as baseline methods for comparison, including relevance based recommendation [1–3,8], diversity query recommendation [4,9] and utility based recommendation [11]. We defined four evaluation metrics to evaluate the results of different methods. The experiment results have shown that QRM outperformed all other methods in all evaluations.

The remainder of this paper is organized as follows. Section 2 gives basic definitions on data representation and the problem statement. Models and solutions to rank candidate queries for recommendation are introduced in Sect. 3. Section 4 presents experiments and results, including real life data sets, performance evaluation matrices, baseline techniques for comparisons and result discussions. Finally, conclusions and future research directions are given in Sect. 5.

2 Preliminaries

We first define query cycle as a primary data representation.

Definition 1. *A query cycle is a 5-tuple* $QC =< UID, T, Q, C, U >$, *where* UID *denotes a user identifier,* T *denotes a time stamp,* Q *denotes a query,* C *denotes a state of query result: either "clicked" or "un-clicked",* U *denotes a set of clicked URLs.*

A query log data can be considered as a set of query cycle records indexed by UID and T. A Web search task is usually carried out through a sequence of query cycles called a search session.

Definition 2. *A search session* SS *is a sequence of query cycles*

$$< QC_0, QC_1, \cdots, QC_i, QC_{i+1}, \cdots, QC_n >$$

where $QC_0.UID = \cdots = QC_n.UID$, $QC_0.T < \cdots < QC_i.T < QC_{i+1}.T < \cdots < QC_n.T$, $QC_{i+1}.T - QC_i.T \leq t_\theta$ *where* t_θ *is a given time threshold.*

The first query of a search session is most important because it reflects what information the user wants to find in the search task. We call the first query as the initial query.

Definition 3. *The initial query* q_l *of a search session* SS_l *is in the first query cycle* QC_0.

Given a time interval threshold t_θ, we can easily extract all search sessions from a query log file and reorganize the log file in search sessions. The search sessions with the same initial query form a search session group.

Definition 4. *A search session group SSG_l consists of a group of N_l search sessions which have the same initial query q_l. The first query cycles QC_0 of all search sessions are not included in SSG_l.*

Definition 5. *Given an initial query q_l, the set of all distinct queries in search session group SSG_l are defined as the candidate queries of q_l, excluding q_l itself.*

Based on the above definitions, we formulate Web query recommendation as a candidate query ranking problem. Let $\boldsymbol{Q}$ denote a set of initial queries from a query log data and $\boldsymbol{Q_l} = \{q_{l,1}, \cdots, q_{l,T_l}\}$ denote a set of T_l candidate queries to $q_l \in \boldsymbol{Q}$. Given q_l and $\boldsymbol{Q_l}$, we rank the candidate queries in $\boldsymbol{Q_l}$ and select the top K queries to recommend to the user.

3 Query Ranking Model and Solutions

3.1 Query Ranking Model

Let Q be a random variable to index candidate queries in $\boldsymbol{Q}_l$, $C = \{unclick = 0, click = 1\}$ a random variable indicating whether the user clicks the return result of candidate query $q_{l,t}$ or not, and $S = \{unsatisfied = 0, satisfied = 1\}$ a random variable indicating whether the user is satisfied with the clicked document or not. Given that the user has issued an initial query q_l, we can use the joint probability $P(Q = q_{l,t}, C = 1, S = 1|q_l, q_{l,t} \in \boldsymbol{Q}_l)$ to measure the potential usefulness of a candidate query $q_{l,t}$ to the user. The higher the probability, the more useful the candidate query $q_{l,t}$ to the user. To simplify the computation, we make an assumption that the conditional probabilities of three random variables are independent. Therefore, we have

$$\begin{aligned} &P(Q = q_{l,t}, C = 1, S = 1|q_l, q_{l,t}) \\ =&P(Q = q_{l,t}|q_l, q_{l,t}) \cdot P(C = 1|Q = q_{l,t}, q_l, q_{l,t}) \cdot P(S = 1|C = 1, q_l, q_{l,t}) \end{aligned} \tag{1}$$

Since it is difficult to compute the three conditional probabilities, we define three utility measures to indirectly compute them.

Definition 6. *Given an initial query q_l and a candidate query $q_{l,t} \in \boldsymbol{Q}_l$, perceived utility $\alpha_{l,t}$ under q_l and $q_{l,t}$ is defined as the probability that user will click the query result of $q_{l,t}$.*

Definition 7. *Given an initial query q_l and a candidate query $q_{l,t} \in \boldsymbol{Q}_l$, posterior utility $\beta_{l,t}$ under q_l and $q_{l,t}$ is defined as the information gain that the user obtains from the clicked result.*

Definition 8. *Given an initial query q_l and a candidate query $q_{l,t} \in \boldsymbol{Q}_l$, the query-level utility $\gamma_{l,t}$ under q_l and $q_{l,t}$ is defined as the attractiveness of $q_{l,t}$ to the user.*

Given the above definitions, we note the following relations: $\alpha_{l,t} = P(C = 1|Q = q_{l,t}, q_l, q_{l,t})$, $\beta_{l,t} \propto P(S = 1|C = 1, q_l, q_{l,t})$ and $\gamma_{l,t} \propto P(Q = q_{l,t}|q_l, q_{l,t})$. Therefore, we have $P(Q = q_{l,t}, C = 1, S = 1|q_l, q_{l,t}) \propto \alpha_{l,t} * \beta_{l,t} * \gamma_{l,t}$. Since ranking is relative, we can use $\alpha_{l,t} * \beta_{l,t} * \gamma_{l,t}$ to rank candidate queries in $\boldsymbol{Q}_l$, which is equivalent to using the joint probability (1).

3.2 Search Session Satisfaction State

Given an initial query q_l and search session group SSG_l, we first determine the relevance of each candidate query $q_{l,t}$ to q_l. We extract all queries $q_{l,t}$, q_l and their clicked URLs from SSG_l and QC_0 of all $SS_{l,j} \in SSG_l$, and then create a bipartite graph in which the edge between a query and a URL indicates that the query and the URL occur in a query cycle. Let $U(i)$ be the set of URLs connected to query node i in the bipartite graph. We construct a new graph $G = (V, E)$ where $V = \boldsymbol{Q_l} \cup \{q_l\}$ and $E = \{e(i, j)|U(i) \cap U(j) \neq \emptyset, i, j \in V\}$. On G, if a query node has a path to the initial query q_l, we say that the query is relevant to q_l. We use $O_{l,j,i} = 1$ to indicate that the query in query cycle i of search session j is relevant to q_l. Otherwise, $O_{l,j,i} = 0$

For each search session $SS_{l,j}$ in SSG_l, we compute the smoothed total number of clicked URLs $\theta_{l,j}$ for $SS_{l,j}$ as

$$\theta_{l,j} = \sigma(\sum_{i=1}^{M_{l,j}} I(O_{l,j,i} = 1) \cdot |U_{l,j,i}|), \tag{2}$$

where $M_{l,j}$ is the number of query cycles in $SS_{l,j}$, $|U_{l,j,i}|$ is the number of clicked URLs in query cycle i and $\sigma(x)$ is a monotonic smoothing function of

$$\sigma(x) = \frac{1}{1 + exp(-x)} \tag{3}$$

The average of $\theta_{l,j}$ is

$$\bar{\theta}_l = \frac{\sum_{j=1}^{N_l} \theta_{l,j}}{N_l}. \tag{4}$$

Using $\bar{\theta}_l$ as a threshold, if $\theta_{l,j} \geq \bar{\theta}_l$, we say the user was satisfied with $SS_{l,j}$ and write $S_{l,j} = 1$. Otherwise, $S_{l,j} = 0$ if $\theta_{l,j} < \bar{\theta}_l$.

3.3 Perceived Utility $\boldsymbol{\alpha}$

Given an initial query q_l and the set of candidate queries $\boldsymbol{Q}_l$ and search session group SSG_l, perceived utility $\alpha_{l,t}$ is estimated as

$$\alpha_{l,t} = \frac{\sum_{j=1}^{N_l} \sum_{i=1}^{M_{l,j}} I(Q_{l,j,i} = q_{l,t}) \cdot I(C_{l,j,i} = 1)}{\sum_{j=1}^{N_l} \sum_{i=1}^{M_{l,j}} I(Q_{l,j,i} = q_{l,t})}, \tag{5}$$

where $Q_{l,j,i}$ is the query in query cycle i of $SS_{l,j}$, $C_{l,j,i} = 1$ indicates that the result is clicked in query cycle i of $SS_{l,j}$, $I(.)$ is an indicator function, $M_{l,j}$ is the number of query cycles in $SS_{l,j}$ and N_l is the number of search sessions in SSG_l. The larger the proportion of the results from candidate query $q_{l,t}$ which were clicked, the higher the perceived utility $\alpha_{l,t}$ for query $q_{l,t}$, and the more potentially useful the candidate query $q_{l,t}$ to the user.

3.4 Posterior Utility $\boldsymbol{\beta}$

Posterior utility $\beta_{l,t}$ for candidate query $q_{l,t}$ is defined as the information gain that the user obtain from the clicked result. Given a search session $SS_{l,j}$, let $\beta_{l,j,k}$ be the information gain obtained from query cycle k. The total information gain accumulated up to query cycle i in $SS_{l,j}$ is $\sum_{k=1}^{i} (I(C_{l,j,k} = 1) \cdot I(O_{l,j,k} = 1) \cdot \beta_{l,j,k})$, The probability that the user is satisfied up to query cycle i is defined as

$$P(S_{l,j,i} = 1|\boldsymbol{C_{l,j,1:i}}, \boldsymbol{O_{l,j,1:i}}) = \sigma(\sum_{k=1}^{i} (I(C_{l,j,k} = 1) \cdot I(O_{l,j,k} = 1) \cdot \beta_{l,j,k})), \quad (6)$$

where $\boldsymbol{C_{l,j,1:i}}$ and $\boldsymbol{O_{l,j,1:i}}$ are two vectors of 0 or 1, indicating the states of clicks and relevance of queries and $\sigma(x)$ is the smooth function (3).

Since $S_{l,j,i}$ has only two states, we have

$$P(S_{l,j,i} = 0|\boldsymbol{C_{l,j,1:i}}, \boldsymbol{O_{l,j,1:i}}) = 1 - P(S_{l,j,i} = 1|\boldsymbol{C_{l,j,1:i}}, \boldsymbol{O_{l,j,1:i}}), \quad (7)$$

The probability for occurrence of search session group SSG_l is

$$\prod_{j=1}^{N_l} \prod_{i=1}^{M_{l,j}} P(S_{l,j,i}|\boldsymbol{C_{l,j,1:i}}, \boldsymbol{O_{l,j,1:i}})^{S_{l,j,i}} \cdot (1 - P(S_{l,j,i}|\boldsymbol{C_{l,j,1:i}}, \boldsymbol{O_{l,j,1:i}}))^{1-S_{l,j,i}}, \quad (8)$$

We take the logarithm of (8) and maximize $\mathcal{L}(\boldsymbol{\beta_l})$ with constraints $\boldsymbol{\beta_l} \geq \mathbf{0}$ by optimizing the objective function

$$\Lambda(\boldsymbol{\beta_l}) = \mathcal{L}(\boldsymbol{\beta_l}) + \sum_{t=1}^{T_l} \lambda_t \cdot \beta_{l,t} - \mu_\beta \|\boldsymbol{\beta_l}\|_2, \quad (9)$$

subject to

$$\beta_{l,t} \geq 0 (1 \leq l \leq L, 1 \leq t \leq T_l), \quad (10)$$

$$\lambda_t \geq 0 (1 \leq t \leq T_l), \quad (11)$$

$$\lambda_t \cdot \beta_{l,t} = 0 (1 \leq l \leq L, 1 \leq t \leq T_l), \quad (12)$$

where λ_t are Lagrangian coefficients and μ_β is the regularization parameter. The Eqs. (10), (11) and (12) are Karush-Kuhn-Tucker (KKT) optimality conditions. And we also impose a L2-norm constraint on the objective function to avoid singular solutions.

As in [10], we convert the inequality constraints $\boldsymbol{\beta_l} \geq \mathbf{0}$ to equality constraints by introducing slack variables z_t $(1 \leq t \leq T_l)$ which satisfy $\beta_{l,t} - z_t = 0$, where $z_t \geq 0$. Due to the fact that λ_t and z_t must be positive or zero, we put all λ_t and z_t in a quadratic form, i.e., λ_t^2 and z_t^2.

Applying these steps, we get the following transformed optimality conditions:

$$\begin{cases} \dfrac{\partial}{\partial \beta_{l,t}} (\mathcal{L}(\boldsymbol{\beta_l}) + \sum_{t=1}^{m} \lambda_t^2 \beta_{l,t} - \mu_\beta \|\boldsymbol{\beta_l}\|_2) = 0, \\ -\beta_{l,t} + z_t^2 = 0, \\ \lambda_t^2 \beta_{l,t} = 0. \end{cases} \quad (13)$$

Using Newton-Raphson method, we can solve (13) by iteratively solving following equations to obtain all $\boldsymbol{\beta_l}$

$$\begin{cases} \beta_{l,t}(T) = \beta_{l,t}(T-1) + \Delta\beta_{l,t}(T-1) \\ z_t(T) = z_t(T-1) + \Delta z_t(T-1) \\ \lambda_t(T) = \lambda_t(T-1) + \Delta\lambda_t(T-1) \end{cases} \tag{14}$$

where T is the iteration number and

$$\begin{cases} \Delta\beta_{l,t} = \dfrac{-\beta_{l,t}\frac{\partial\mathcal{L}(\boldsymbol{\beta_l})}{\partial\beta_{l,t}} + 2\mu_\beta\beta_{l,t}^2}{\beta_{l,t}\frac{\partial^2\mathcal{L}(\boldsymbol{\beta_l})}{\partial\beta_{l,t}^2} - 2\mu_\beta\beta_{l,t} - \lambda_t^2}, \\ \Delta z_t = \dfrac{\beta_{l,t} - z_t^2 + \Delta\beta_{l,t}}{2z_t}, \\ \Delta\lambda_t = \dfrac{-\lambda_t^2\beta_{l,t} - \lambda_t^2\Delta\beta_{l,t}}{2\lambda_t\beta_{l,t}}. \end{cases} \tag{15}$$

3.5 Query-Level Utility γ

Query-level utility is defined as attractiveness of a candidate query to the user, and it affects the user's behavior of carrying out search sessions. Given a search session group SSG_l, let $\boldsymbol{Q_l}' \subset \boldsymbol{Q_l}$ be the subset of candidate queries that occur in the first $(i-1)$ query cycles of $SS_{l,j}$. The probability that candidate query $q_{l,t}$ is reformulated by the user in query cycle i of $SS_{l,j}$ is defined as

$$P(Q_{l,j,i} = q_{l,t}|\boldsymbol{Q_l}') = \frac{I(q_{l,t} \in \boldsymbol{Q_l}') \cdot \gamma_{l,t}}{\sum_{q_{l,*} \in \boldsymbol{Q_l}'} \gamma_{l,*}} + \frac{I(q_{l,t} \in (\boldsymbol{Q_l} - \boldsymbol{Q_l}')) \cdot \gamma_{l,t}}{\sum_{q_{l,*} \in (\boldsymbol{Q_l} - \boldsymbol{Q_l}')} \gamma_{l,*}}, \tag{16}$$

where $\gamma_{l,t}$ is the query-level utility of $q_{l,t}$, $I(.)$ is an indicator function, (*) spans $\boldsymbol{Q_l}$.

The probability for occurrence of search session group SSG_l is

$$\prod_{j=1}^{N_l}\prod_{i=1}^{M_{l,j}} P(Q_{l,j,i}|\boldsymbol{Q_l}'), \tag{17}$$

We take the logarithm of the likelihood function (17) and obtain

$$\mathcal{L}(\boldsymbol{\gamma_l}) = \sum_{j=1}^{N_l}\sum_{i=1}^{M_{l,j}} \log(\frac{I(q_{l,t} \in \boldsymbol{Q_l}') \cdot \gamma_{l,t}}{\sum_{q_{l,*} \in \boldsymbol{Q_l}'} \gamma_{l,*}} + \frac{I(q_{l,t} \in (\boldsymbol{Q_l} - \boldsymbol{Q_l}')) \cdot \gamma_{l,t}}{\sum_{q_{l,*} \in (\boldsymbol{Q_l} - \boldsymbol{Q_l}')} \gamma_{l,*}}), \tag{18}$$

The solution to (18) is similar to solving $\boldsymbol{\beta_l}$ except that we replace $\beta_{l,t}$ with $\gamma_{l,t}$ in (14) and (15) and substitute $\frac{\partial\mathcal{L}(\boldsymbol{\gamma_l})}{\partial\gamma_{l,t}}$ to $\frac{\partial\mathcal{L}(\boldsymbol{\beta_l})}{\partial\beta_{l,t}}$ whereas $\frac{\partial^2\mathcal{L}(\boldsymbol{\gamma_l})}{\partial\gamma_{l,t}^2}$ to $\frac{\partial^2\mathcal{L}(\boldsymbol{\beta_l})}{\partial\beta_{l,t}^2}$ in (15).

4 Experiments

The real query log data 'Spring 2006 Data Asset' from Microsoft Research[4] was used in the experiments. We removed symbols such as "?", "#", "+", etc. and converted capital letters to lower cases. We found that the log data had about 6.5 million unique queries and 4.9 million unique URLs. We reorganized the log data into search sessions and search session groups. There were about 3 million sessions and 1.8 million SSGs. We treat the initial queries of SSGs as test queries.

4.1 Baseline Methods

We selected six baseline methods in three categories for comparison.

relevance based query recommendation:

- Adjacency (ADJ): Given a test query q, the top K frequent queries in the same search session adjacent to q are recommended to the user [7].
- Co-occurrence (COO): Given a test query q, the top K frequent queries occurred in the same search session with q are recommended to the user [5].
- Query-Flow Graph (QFG): In this method, a query-flow graph is first constructed based on the queries appearing in succession in the same search session, and a random walk is performed on this graph for query recommendation [2].

diversity query recommendation:

- Hitting Time(HT): In this method, a query relation graph is constructed from a query-URL bipartite graph. A random walk is conducted on the graph and the hitting times are used to select queries [9].

utility based query recommendation:

- Query Utility Model(QUM): This is a utility query recommendation method which is focussed on document-level utility [11].
- Document-Level Utility(DLU): To evaluate the benefit of query-level utility, we defined a document-level utility only(use $\alpha * \beta$ to recommend queries) method as a baseline method for comparison.

4.2 Evaluation Metrics

To evaluate the recommendation results, we look into both query-level utility and document level utility. The query-level utility affects the attractiveness of a recommended query to the user whereas the document-level utility affects the relevance of the recommended query result to the initial query. We use relative length and lexical similarity of recommended queries to evaluate their query-level

[4] http://research.microsoft.com/users/nickcr/wscd09/

utilities and use QRR and MRD introduced in [11] to evaluate their document-level utilities. Given the test query q_l, four evaluation metrics are defined as

$$RLQRR(q_{l,t}) = RL(q_{l,t}) * QRR(q_{l,t}) \tag{19}$$

$$RLMRD(q_{l,t}) = RL(q_{l,t}) * MRD(q_{l,t}) \tag{20}$$

$$LSQRR(q_{l,t}) = LS(q_{l,t}) * QRR(q_{l,t}) \tag{21}$$

$$LSMRD(q_{l,t}) = LS(q_{l,t}) * MRD(q_{l,t}) \tag{22}$$

where

- $RL(q_{l,t})$ measures the relative length of $q_{l,t}$ computed as $RL(q_{l,t}) = \frac{|\omega(q_l)|}{|\omega(q_{l,t})|}$ where $q_{l,t}$ is the recommended query and $\omega(.)$ denotes the set of terms in a query.
- $LS(q_{l,t})$ measures lexical similarity of two queries as $LS(q_{l,t}) = \frac{\boldsymbol{q_l} \cdot \boldsymbol{q_{l,t}}}{\|\boldsymbol{q_l}\|\|\boldsymbol{q_{l,t}}\|}$ where $\boldsymbol{q}$ represents a term vector of a query.
- $QRR(q_{l,t})$ measures the query relevant ratio as

$$QRR(q_{l,t}) = \frac{\sum_{j=1}^{N_l} \sum_{i=1}^{M_{l,j}} I(R_{l,j,i} > 0) \cdot I(Q_{l,j,i} = q_{l,t})}{\sum_{j=1}^{N_l} \sum_{i=1}^{M_{l,j}} I(Q_{l,j,i} = q_{l,t})}, \tag{23}$$

- $MRD(q_{l,t})$ measures the mean relevant document as

$$MRD(q_{l,t}) = \frac{\sum_{j=1}^{N_l} \sum_{i=1}^{M_{l,j}} R_{l,j,i} \cdot I(Q_{l,j,i} = q_{l,t})}{\sum_{j=1}^{N_l} \sum_{i=1}^{M_{l,j}} I(Q_{l,j,i} = q_{l,t})}, \tag{24}$$

  where $R_{l,j,i}$ is the number of clicked relevant documents in query cycle $Q_{l,j,i}$. To compute $R_{l,j,i}$, we need the URLs clicked by recommended queries being labeled as 'relevant' or 'irrelevant'. We totally labeled $25,356$ URLs in the query log.

When we evaluate the performance of QFG, ADJ, COO and HT, the metrics 23 and 24 are extended to whole data set rather than $l - th$ SSG.

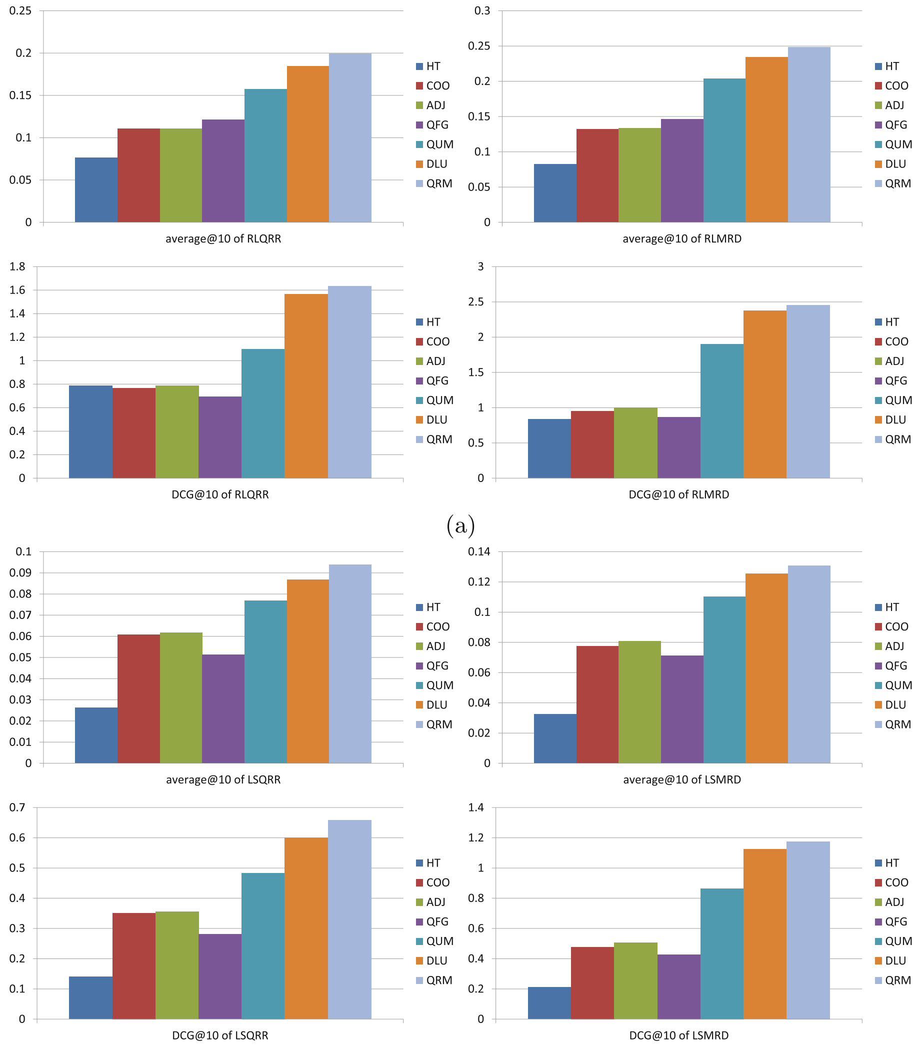
We evaluate the average performance of the top 10 recommended queries (such as $query_{l,t}$) as

$$AVG(q_l, 10) = \frac{\sum_{t=1}^{10} metrics(query_{l,t})}{10}. \tag{25}$$

To consider the ranking position of a recommended query, we use *discounted cumulative gain*[6] to evaluate the top 10 recommended queries queries as

$$DCG(q_l, 10) = \sum_{t=1}^{10} \frac{2^{metrics(query_{l,t})} - 1}{log(t+1)}, \tag{26}$$

where $metrics(.)$ can be any equation from 19 to 22.



(a)

(b)

Fig. 1. Evaluation on the results of seven methods with the query-level and document-level utility metrics. Sub-figure (a) shows the results with metrics of RLQRR and RLMRD; Sub-figure (b) shows the results with metrics of LSQRR and LSMRD.

4.3 Experiment Results

We ran the seven methods on the whole data set and randomly selected 100 test queries whose candidate queries are more than 40 but less than 100 and evaluated them with the average value of metrics 25 and 26 which are denoted as *average*@10 and *DCG*@10 respectively. Figure 1 shows the bar charts of the evaluation results. The higher a bar, the better the result.

Table 1. Executing time of seven methods on 600 test queries.

Method	QRM	DLU	QUM	QFG	COO	ADJ	HT
Executing Time(min)	48.56	29.86	22.79	20.63	1.2	1.3	13.92

Since utility based methods can recommend queries with more relevant documents in their search results than that of queries recommended by similarity based methods, the utility based method QRM, DLU and QUM performed better than similarity based method HT, COO, ADJ and QFG as shown in Fig. 1.

QRM performed better than DLU as shown by the right bar because QRM considered document-level utility as well as query-level utility, and can recommend more attractive candidate queries which are more similar to its test query and contain fewer key words, but DLU considered only document-level utility. The comparison between results of QRM and DLU indicate that the additional query-level utility can improve the attractiveness of recommended queries effectively. DLU and QUM are both document-level utility based methods. QUM performed worse than DLU due to that it did not distinguish irrelevant queries and use unrealistic search session satisfaction state estimating method. However, DLU took irrelevant queries into consideration and use a reasonable estimating method and can recommend queries whose search results contain more relevant documents, so it performed better than QUM.

The t-test $(p - value \leq 0.05)$ over the results also showed that the performance improvement of QRM was significant comparing with other baselines.

4.4 Executing Time

We randomly selected 600 test queries and ran each of seven methods to recommend 10 queries for them on a same Linux server. We recorded the executing time of seven methods in Table 1. From Table 1, we can see that the overhead of QRM increases only 1–3 times than other baselines except COO and ADJ. COO and ADJ cost fewest because they just summarize some statistical information from organized search sessions. It is worth noting that, for each test query, QRM recommends candidate queries independently, therefore, we can ran QRM in a parallel computing framework with little executing time.

5 Conclusions

In this paper, we have presented QRM, a model based method for query recommendation in a search engine. In QRM, three levels of information are used in building a ranking model, including the recommended query itself, the user behavior in a search session and the user satisfaction of the search session. We have defined query-level utility, perceived utility and posterior utility to compute the model. Experiment analysis on a real world data set has demonstrated that in comparison with six baseline query recommendation methods, QRM was superior to other methods in selecting candidate queries.

Our future work is to develop an incremental query update method to keep the recommendation queries up to date. Another challenging task is to extract recommendation queries from big query log data. Distributed algorithms need to be developed to carry out this task.

Acknowledgment. This research is supported by Shenzhen New Industry Development Fund under Grant No.JC201005270342A.

References

1. Beeferman, D., Berger, A.: Agglomerative clustering of a search engine query log. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 407–416 (2000)
2. Boldi, P., Bonchi, F., Castillo, C., Donato, D., Gionis, A., Vigna,S.: The query-flow graph: model and applications. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08, pp. 609–618 (2008)
3. Cao, H., Jiang, D., Pei, J., He, Q., Liao, Z., Chen, E., Li, H.: Context-aware query suggestion by mining click-through and session data. In: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 875–883 (2008)
4. Guo, J., Cheng, X., Xu, G., Shen, H.: A structured approach to query recommendation with social annotation data. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10, pp. 619–628 (2010)
5. Huang, C.-K., Chien, L.-F., Oyang, Y.-J.: Relevant term suggestion in interactive web search based on contextual information in query session logs. J. Am. Soc. Inf. Sci. Technol. **54**(7), 638–649 (2003)
6. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf. Syst. **20**(4), 422–446 (2002)
7. Jones, R., Rey, B., Madani, O., Greiner, W.: Generating query substitutions. In: Proceedings of the 15th International Conference on World Wide Web, WWW '06, pp. 387–396 (2006)
8. Li, L., Xu, G., Yang, Z., Dolog, P., Zhang, Y., Kitsuregawa, M.: An efficient approach to suggesting topically related web queries using hidden topic model. World Wide Web **16**, 273–297 (2013)
9. Mei, Q., Zhou, D., Church, K.: Query suggestion using hitting time. In: Proceeding of the 17th ACM Conference on Information and Knowledge Management, pp. 469–477 (2008)
10. Tognola, G., Rainer, B.: Unlimited point algorithm for opf problems. IEEE Trans. Power Syst. **14**(3), 1049–1052 (1999)
11. Zhu, X., Guo, J., Cheng, X., Lan, Y.: More than relevance: High utility query recommendation by mining users's search behaviors. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12, pp. 37–46 (2012)

Toward Mining User Traversal Patterns in the Indoor Environment

Shan-Yun Teng[1], Tzu-Yuan Chung[1], Kun-Ta Chuang[1](✉), and Wei-Shinn Ku[2]

[1] Department of Computer Science and Information Engineering,
National Cheng Kung University, Tainan, Taiwan
{sydang.ncku,tzuyuan.chung}@gmail.com, ktchuang@mail.ncku.edu.tw

[2] Department of Computer Science and Software Engineering, Auburn University,
Auburn, USA
weishinn@auburn.edu

Abstract. We in this paper explore a new mining paradigm, called *Indoor Traversal Patterns* (abbreviated as *ITP*), to discover user traversal behavior in the mall-like indoor environment. The *ITP* algorithm can identify user traversal sequences from uncertain user itineraries with the RFID-based indoor positioning technology. Note that it is a highly challenging issue in the indoor environment to retrieve the precise locations in the indoor environment. Since previous works on mining user moving patterns usually rely on the precise spatiotemporal information from GPS signals, it is difficult to apply similar approaches to discover user traversal behavior in the indoor environment. We therefore develop a framework to transform the RFID antenna data to uncertain user traversal transactions, and further diminish the uncertainty before mining the indoor traversal patterns. Our experimental studies show that the proposed *ITP* algorithm can effectively overcome the impact from location uncertainty and discover high-quality traversal patterns, to provide insightful observation for marketing decision.

1 Introduction

As the growth of modern cities, a significant portion of outdoor activities in human daily life has shifted to indoor activities nowadays. The trend leads to the apparent increase of time spent in some indoor spaces, e.g., shopping malls, for most people, thus driving the applications of discovering the human behavior in the indoor space. The existing algorithms for mining user behavior from user trajectories, such as user moving patterns or the visiting sequences, are generally developed for the outdoor environment [3], in which the precise user locations from GPS signals or cellular positioning can be acquired consecutively. However, due to the privacy concern and the limitation from hardware deployment [17], to retrieve the precise locations is a highly challenging issue in the indoor environment, making the extension of previous algorithms used in the outdoor space needs further justification in the indoor environment [8].

Recent advancements in indoor positioning technologies, such as Wi-Fi, RFID and Bluetooth, enable the development of various location-based services in the

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 677–688, 2014.
DOI: 10.1007/978-3-319-13186-3_60

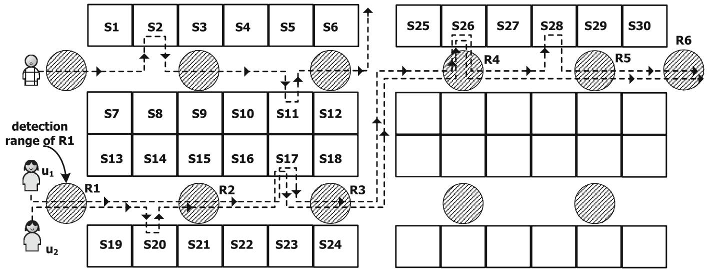


Fig. 1. An illustrative example of indoor traversal pattern.

indoor environment [15]. In particular, RFID is recently highlighted as an important physical media for indoor positioning [17]. The technique has been extensively utilized to track objects for supply chain management, health care, and so on. Generally, RFID readers are deployed in some critical places. The reader will recognize the presence of a user (with a RFID tag) when the user passes the detection range of the reader. Due to the high penetration rate of RFID, it is a practicable means to support marketing decisions for indoor vendors by mining indoor user moving behavior under the RFID positioning media.

However, the RFID-based positioning will incur the uncertainty of user locations. Note that the raw data collected by RFID readers is unreliable due to RF interference, limited detection range and tag orientation. In addition, RFID readers cannot be deployed with the high coverage rate because of the considerable capital cost and privacy concerns [5,13]. The inherent characteristics of location uncertainty is not addressed in previous mining algorithms in the outdoor environment. How to discover user patterns in the uncertain environment is still left unresolved in the literature.

As such, we in this paper study a new paradigm, called *Indoor Traversal Patterns* (abbreviated as *ITP*), in data mining research. Specifically, the *ITP* mining can be used to capture the user shopping behavior in a mall-like RFID-based indoor environment. Figure 1 illustrates an scenario of mining *ITP*, where simulates a floor in the shopping mall. In the example, each store is denoted by S_j, and a set of RFID readers, denoted by R_i, are deployed in the aisle. The shopping itinerary of each user will be partially identified by RFID readers while he/she passes the corresponding detection range. For example, both users u_1 and u_2 are detected by readers $\{R_1, R_2, R_3, R_4, R_5, R_6\}$ successively. Note that a significant portion of user trajectories is inexistent in the RFID-based indoor environment. Moreover, since no reader can be deployed within the room for the privacy issue [17], we cannot ensure whether u_2 enters S_{20} or not. For every user itinerary, we only have the sequence of passed readers and the corresponding temporal information. The uncertain part, including the stay in the store, must be inferred from the spatiotemporal RFID reading list.

In this work, we discuss the algorithm of mining *ITP* from the RFID-based uncertain data set in the indoor environment. We rely on the previous work in [17] to clean RFID raw data and provide precise temporal information of reading sequences. In Sect. 2, we will give related works and Sect. 3 will introduce the mining framework. The experimental results are described in Sect. 4. Finally, this paper concludes with Sect. 5.

2 Related Works

In this section, we review previous works related to frequent trajectory pattern mining and RFID technologies.

Mining sequential patterns from user trajectories: The sequential pattern mining problem was first introduced in [2], which defined the problem over a database D of sequences and presented solutions of retrieving all the frequent sequences in D. Giannotti *et al.* [3] developed an extension of the sequential pattern mining paradigm that analyzes the trajectories of moving objects. They introduced trajectory patterns as concise descriptions of frequent behaviors, in terms of both space and time.

Liu *et al.* [7] proposed to employ RF tag arrays in mining frequent trajectory patterns for activity monitoring. Specifically, they offset the noise of RF tag data and mine frequent trajectory patterns as models of regular activities by developing a practical fault-tolerant method.

However, due to the inherent differences in spatial characteristics, indoor moving pattern mining need different models and cannot directly apply mature techniques from their outdoor counterparts.

RFID-Based Track and Trace: RFID is a very popular electronic tagging technology that allows objects to be automatically identified at a distance using an electromagnetic challenge-and-response exchange of data [12]. An RFID-based system consists of a large number of low-cost tags that are attached to objects, and readers which can identify tags without a direct line-of-sight through RF communications. RFID technologies enable exceptional visibility to support numerous track and trace applications in different fields [16]. However, the raw data collected by RFID readers is inherently noisy and inconsistent [5,10]. Therefore, middleware systems are required to correct readings and provide cleansed data [5]. In addition to the unreliable nature of RFID data streams, another limitation is that due to the high cost of RFID readers, RFID readers are mostly deployed such that they have disjoint activation ranges in the settings of indoor tracking. Furthermore, privacy (i.e., readers are deployed in hallways rather than rooms in office buildings) is also an important concern [13].

To overcome the above limitations, RFID data cleansing is a necessary step to produce consistent data to be utilized by high-level applications. Tran *et al.* [11] used a sampling-based method called particle filtering to infer clean and precise event streams from noisy raw data produced by mobile RFID readers. Three enhancements are proposed in their work to make traditional particle filter techniques scalable. However, their work is mainly designed for warehouse settings where objects remain static on shelves, which is quite different from our setting where objects move around in a building. Therefore, Tran's approach of adapting and applying particle filters cannot be directly applied to our settings.

Another limitation of [11] is that they did not explore further utilization of the output event streams for high-level applications. Ku *et al.* [6] employed a different sampling method called Markov Chain Monte Carlo (MCMC) to infer objects' locations on shelves in warehouses. Their method takes advantage of the spatial and temporal redundancy of raw RFID readings, and also considers environmental constraints such as the capacity of shelves, to make the sampling process more precise. Their work also focuses on warehouse settings; thus is not suitable for our problem of general indoor settings. The works in [9,13] target settings such as office buildings, which are similar to our problem. They use particle filters in their preprocessing module to generate probabilistic streams, on which complex event queries such as "Is Joe meeting with Mary in Room 203?" can be processed. However, their goal is to answer event queries instead of frequent trajectory pattern mining, which is different from the goal of this research. Furthermore, a hot research topic of the robotics research community, simultaneous localization and mapping (SLAM), also makes extensive utilization of particle filters [14].

Lu et al. [8] captured the tracking data of users in indoor spaces to identify typical movements behavior among objects. They utilize the indoor topology and apply traditional frequent sequential patterns. In the work, readers are deployed side by side in a defined region, so that they can get the precise region corresponding to the location of the user. In such situations, their trajectories are in fact precise and not uncertain. However, considering the privacy of users, it is generally required that readers should be put sparsely on the aisle instead of rooms. The technique of sequential pattern mining in this paper dose not work in the case of location uncertainty. It is orthogonal to our work focusing on handling trajectories with uncertain location information.

3 The ITP Framework

In this section, we describe our framework to discover *indoor traversal patterns*. We first introduce the definition of *indoor traversal patterns* in Sect. 3.1, and the system framework will be presented in Sect. 3.2.

3.1 Problem Definition

We give the necessary definitions as follows.

Definition 1 ITE (Indoor Traversal Event): *Suppose that user u_i is detected by the RFID reader$r_j$starting from time t_{sj} , and r_j continues sensing the appearance of u_i until time t_{ej}. Then u_i keep being detected by reader r_k, starting from time t_{sk} and ending with t_{ek}. We have an indoor traversal event of u_i, which is denoted by the 4-tuple $(u_i, r_j, r_k, [t_{ej}, t_{sk}])$. And $t(e_i)$ denotes the time interval from r_j to r_k.*

Definition 2 UTP (User Indoor Traversal Path): *Suppose that user u_i is successively detected by $r_{1,}$ $r_{2,}$ ..., r_m, where r_m is the last reader in the system*

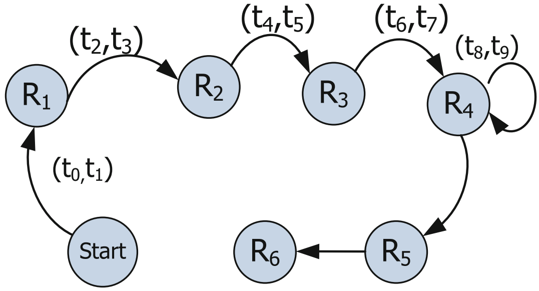


Fig. 2. An illustrative example of indoor traversal path.

that senses the appearance of u_i. The corresponding user indoor traversal path is denoted by $p_i = \{e_1, e_2, ..., e_n\}$, where e_k is the indoor traversal event, for $1 \leq n \leq m$.

An illustrated example of user u_2 is shown in Fig. 2. For ease of presentation, we ignore the case that u_i stays in a location for a long time, longer than the max interval in the system. For such cases, we could treat it as another path.

In addition, following the same principle which is generally used in previous works, we also assume no reader is deployed in the room space due to the privacy issue [17]. However, it is relatively easy to recognize if a user has a valid visit (or windows shopping) in a room. A store can identify the visited time by their experience. For example, a cloth shop can tell that its valid customer should stay within the store longer than 3 min. As such, we give the following definition.

Definition 3 (Steady State In Rooms): *For a region of room S_l, we can define the time stayed in S_l is called valid, i.e., $t_{stay}(S_l) = [t_{min}(S_l), t_{max}(S_l)]$, where $t_{min}(S_l)$ and $t_{max}(S_l)$ are the minimum time and maximum time that we can say a user stays in S_l, respectively. In general, $t_{max}(S_l)$ can be defined as infinite.*

Note that we assume readers are deployed in the road path, meaning that no valid stay in a room for the indoor traversal event e_k.

Definition 4 UVR (Uncertain Visited Rooms according to Indoor Traversal Event): *Suppose that the indoor traversal path $p_i = \{e_1, e_2, ..., e_m\}$, we can transform each successive traversal event e_i to the set of uncertain visited rooms. That is $UVR(e_i) = \{S_{i,1}, S_{i,2}...S_{i,k}\}$.*

Definition 5 UVT (Uncertain Visited Transaction): *Given the set of uncertain visited rooms between any successive traversal events in the indoor traversal path p_i. We can completely transform the RFID antenna data to the transaction of user uncertain visited rooms tr_i, i.e.,*

$$tr_i =< \{S_{1,1}, S_{1,2}...S_{1,k1}\}, ..., \{S_{m,1}, S_{m,2}...S_{m,km}\} > .$$

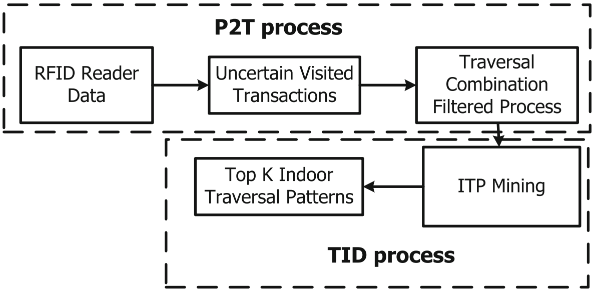

P2T process
RFID Reader Data
Uncertain Visited Transactions
Traversal Combination Filtered Process
ITP Mining
Top K Indoor Traversal Patterns
TID process

Fig. 3. Overview of the ITP framework.

<u>**Problem Formulation**</u> **(Top K Indoor Traversal Patterns Discovery):** *Suppose that a pattern of ITP (indoor traversal pattern) is defined as the form:*

$$< S_{1,1}, S_{1,2}, ..., S_{f,mf} >,$$

and its support value equals to its occurrence count in UVTs*. Given the database of uncertain visited transactions D and the desired number of patterns k, the goal of the ITP framework is to discover top K indoor patterns (abbreviated as TKP) from* UVTs*, according to the support of each ITP.* ■

3.2 System Framework

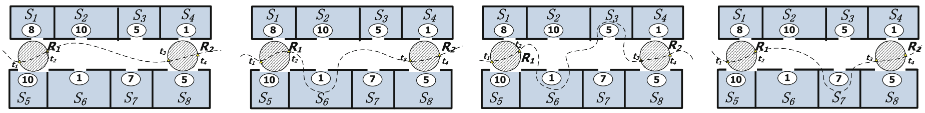
The *ITP* framework is devised to effectively resolve the issue of uncertainty in the visited transactions, and also to efficiently discover the top k indoor traversal patterns. The two core steps of *ITP* framework is shown in Fig. 3. The step one is to transform the *UTPs* in Definition 2 to *UVTs* in Definition 5. The step is named as Path to Transactions (abbreviated as *P2T*). The second step is to efficiently mine *TKPs* (top k indoor traversal patterns) from *UVTs*, and we named the step as top k *ITPs* discovery (abbreviated as *TID*).

The details of these two steps are introduced in the following.

Step 1 : *P2T* (Path to Transaction)

The indoor environment is divided into disjoint rooms, and a set of positioning RFID readers is deployed in the aisle. Accordingly, we have a mapping $E2S : E \rightarrow S$, that maps an event with corresponding readers to their rooms. Given an event e_i with corresponding readers r_j, r_k, a region between r_j, and r_k can be regarded as the set of rooms. $E2S(e_i)$ thus returns the room set $UVR(e_i) = \{S_{i,1}, S_{i,2}...S_{i,k}\}$ as we described in Definition 4.

With the big uncertainty in transforming the *UTPs* to *UVTs*, the returned room set $UVR(e_i)$ could produce several traversal combinations, but only one of them is the correct traversal transaction. So we need a traversal combination filtering process (abbreviated as *TCF*). In the process of combination filtering, we make some rules of matching normal user behavior to filter the traversal combinations with the low probability. Figure 4 shows four examples of possible



(a) Visit no stores (b) Visit one store 6 (c) Visit store 3 and 6 (d) Visit one store 7

Fig. 4. The examples of possible combination of visited stores.

traversal combinations. In each case, there are four rooms S_2, S_3, S_6, S_7 on the path from R_1 to R_2, and each room has a $t_{stay}(S_l)$ as we described in Definition 3. We then describe how we filter out the less possible traversal combinations by using this four cases.

The case shown in Fig. 4(a) is obviously filtered out at first. The goal of our framework is to find *TKPs* in room set form. And there is no any room information we can get from the traversal combination, and so it is not necessary to keep such cases.

Besides, we assume that we have a time interval of 8 min between two successive detections. Since it is generally expected that users will not stay awhile in the road path, the case shown in Fig. 4(b) with low probability can be filtered out comparing to the case shown in Fig. 4(c) during the *P2T* processing.

Algorithm 1. P2T algorithm

```
Require: The User Indoor Traversal Path, i.e., $p_i$;
Ensure: Uncertain Visited Transactions, i.e., $t_r$;
 1: procedure P2T($p_i$)
 2:     $t := NULL$;
 3:     $tr :=<>$;
 4:     for each event $e \in p_i$ do
 5:         $s := E2S(e)$;
 6:         $t :=$ all combinations of $s$;
 7:         for each $t_i \in t$ do
 8:             filter the less possible $t_i$ according to $t_{stay}(e)$ by using $TCF$;
 9:         return $t_i$;
10:         $tr$.concat($t_i$);
11:     return $tr$;
```

In addition, it is also reasonable that users tend to visit few stores in a short time. As such, given a time interval of 8 min, the case in Fig. 4(c) can have a probability smaller than the case in Fig. 4(d). Finally, a filtered process is provided before executing the *TID* processing.

As the above we discussed, it could be inferred that given an *UTP* and the mapping *E2S*, the $UVR(e_i)$ are available to produce an *UVT*. Algorithm 1 describes the process. It takes user indoor traversal path p as input and returns uncertain visited transactions tr. The for-loop (lines 4–11) is iterated to generate tr. It generates traversal combinations from the a indoor traversal event e

(lines 5–6) by given a room set $UVR(e_i)$, and filter the low possible ones according to the time interval $t(e_i)$ (lines 7–9). Furthermore, We name the traversal combination filtering process as *TCF*. Finally, we concatenate all returned t_i as *tr* and compute *tr* as the output (lines 10–11). The algorithm can be used to produce *UVTs*.

Step 2 : *TID* (Top k *ITPs* Discovery)

We extend the Apriori algorithm [1] for mining *TKPs* from *UVTs*. Algorithm 2 describes the process. It takes the number of transactions $|D|$ and the top *tk* as input and returns *TKPs*. The for-loop (lines 3–13) is iterated to generate TKPs. It computes 1-item set first (line 2). Then it generates all candidates of C_{k+1} (the candidate itemset of size $k+1$) according to L_k (the itemset of size k), and save all candidates of C_{k+1} in L_{k+1} and the support of each candidate contained in transaction t in S (lines 4–8). Finally, we choose the top k candidates according to their support and return *TKPs* (line 9–13). The algorithm can be used to produce *TKPs* from *UVTs*.

Algorithm 2. TID algorithm

Desc.: C_k:the candidate itemset of size k; L_k:the itemset of size k;
Input: $|D|$: the number of transactions; *tk*:the top k;
Output: TKP:the top k indoor traversal patterns;

```
 1: procedure TID($|D|$,$tk$)
 2:     $L_1 := \{items\}$;
 3:     for ($k = 1; L_k! = \emptyset; k{+}{+}$) do
 4:         $C_{k+1}$:= candidates generated from $L_k$;
 5:         for ($t = 1; t <= |D|; t{+}{+}$) do
 6:             increment the count of all candidates in $C_{k+1}$ that are contained in
                    transaction $t$;
 7:         $L_{k+1}$:= candidates in $C_{k+1}$;
 8:         $S_{k+1}$:= support of each candidate in $L_{k+1}$;
 9:     for ($x = 1; x <= tk; x{+}{+}$) do
10:         $i$:=the candidate with the biggest value in $S$;
11:         $TKP := i \cup TKP$
12:         $remove(i)$;
13:     return $TKP$
```

In the future, we can also consider other heuristic methodologies to distinguish visited list into the certain part or the uncertain part. For example, we may apply sigmoid function in [4] to achieve this goal during the execution of the Apriori algorithm, and then we get the set of frequent indoor traversal patterns. It is worth mentioning that the transformation from RFID antenna data to uncertain visited transactions will need the data cleaning for the raw data. This effort can be achieved by the implementation of previous work in [17]. The details flow of the framework is depicted in Fig. 3.

4 Experimental Results

We in this section present our experimental studies. The *ITP* algorithm, including both *TID* and *P2T*, are implemented in Java. All the experiments are executed on a 3.40 GHz Core i7 machine with 4 gigabytes of main memory, running on Windows 7 operating system.

4.1 Experimental Environment

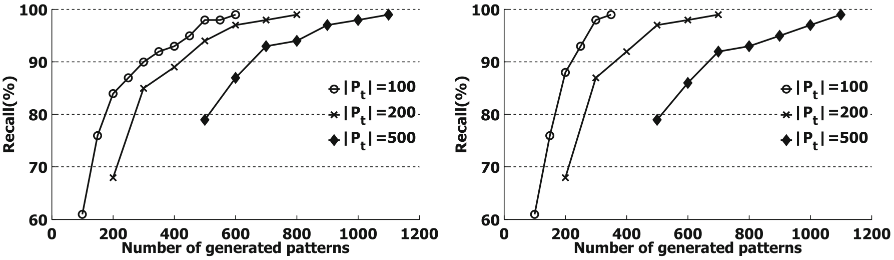
Synthetic datasets are generated by using the data generator which was provided from the authors of [17]. The generator, running on the Linux operating system, can simulate the user walking behavior in the indoor environment with the predefined room layout. The RFID readers, by default, are generally placed on the aisle. We apply a layout of a underground market and simulate normal customer purchasing behavior in the indoor environment. In the underground market, we have 168 stores that sell several kinds of stuffs (i.e., clothes, food, drinks, shoes..., and so on), and randomly placed 50 RFID readers on the aisle. In addition, for a user itinerary, a dwell time is randomly assigned to behave the visiting time period stayed in a store. The distribution of the dwell time in a store follows the normal distribution, and the decision of a user entering into a store or not also follows the normal distribution. Besides, we can set the parameter for the number of customers to generate different amount of transactions. As such, we finally generate a set of user trajectories, that is treated as the ground truth data with the precise location in the indoor market. We further transform the ground truth data to corresponding reader data, and so that we have both ground truth and reader data for performance evaluation.

4.2 Result of Indoor Traversal Patterns Mining

In this section we evaluate the recall rate of generated *ITPs* (abbreviated as p_g) comparing to the ground truth *ITP*(abbreviated as p_t). First, we get *ITPs* from transforming reader data, and we fix the number of D to 10,000 and 20,000. The number of p_t is varied among 100, 200 and 500. Here the recall rate is defined as

$$\frac{|p_g \cap p_t|}{|p_t|}.$$

The results on the recall of frequent *ITPs* are shown in Fig. 5(a) and (b). As we can see, results in both $\mid D \mid = 10{,}000$ or $\mid D \mid = 20{,}000$ have not only the similar increasing curve lines, but also the similar recall at the start point, when the number of p_t equals to 100, 200 and 500. In addition, the recall rate is monotonically increasing as the number of generated patterns increases. It shows two important points. The first one is that when we generate the same number of p_g as p_t, the recall we get will stay in a certainly small range around a specific percentage value, and the number of transactions does not have the obvious impact on the recall rate. The other one is that the growing rate of the



(a) The recall of the ITPs ($|D|$=10,000) (b) The recall of the ITPs.($|D|$=20,000)

Fig. 5. The recall of the generated ITPs.

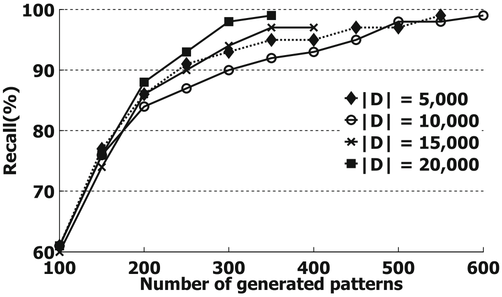


Fig. 6. The recall of the ITPs.

recall is similar when they are in the same condition of the number of p_t. It is clear to see that the proposed *ITP* algorithm can retrieve high-quality result in the uncertain environment.

Furthermore, we fix the number of p_t to 100, and then vary the number of D among 5,000, 10,000, 15,000 and 20,000. The results on the recall of frequent *ITPs* are shown in Fig. 6. We can observe that as the size of D decreases, we have to generate more *ITPs* to reach the recall value of 99 %. In addition, while the size of D is small, we need to generate more *ITPs* to retrieve all *ITPs* discovered in ground truth data. The reason is that in the *TID* process, we could miss couples of items in the whole transactions, which would directly effect the support of items and the consequence of frequent *ITPs*. While the size of D is small, the missing items will obviously impact the result, as compared to the cases when the size of D is huge.

Third, we fix the number of D to 20,000, and then vary the number of p_t among 100, 200, and 500. The results on the recall of frequent *ITPs*, and the difference of recalls with respect to different number of p_g are shown in Fig. 7. It is clearly that for each curve line of the recall of *ITPs*, they are all increase faster at first than in the end, which we can see from the curve of the difference of recall. Besides the recall of the start point are all over 60 percentage. It shows that most of the p_t will be found out at first, then as the p_g increases, the more p_g we have, the less difference of percentage of p_t we get.

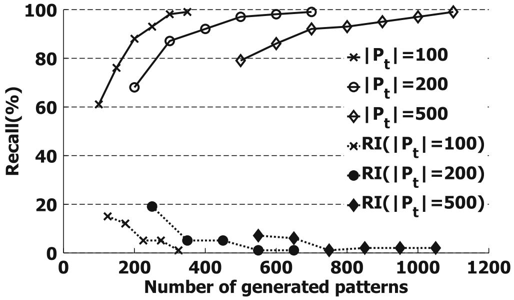
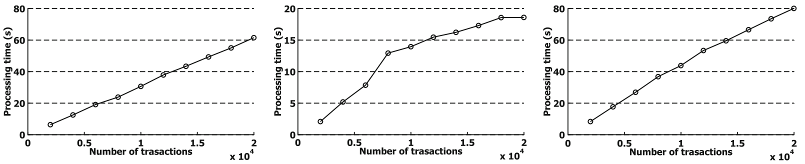


Fig. 7. The increasing recall of the generated ITPs.



(a) P2T execution time (b) TID execution time (c) Total execution time

Fig. 8. The execution time of P2T, TID and total process.

4.3 Execution Time Analysis

In this section we discuss the execution time. We vary the size of D between 2,000 and 20,000, and compute the execution time of the *P2T* and *TID* processing, which we mentioned in the previous section, and the entire processing time.

First, we can see that the result of *P2T* processing time is presented as a perfectly linear increasing curve, as shown in Fig. 8(a). In addition, we report on the mining efficiency of *TID* in Fig. 8(b). It is clear that for a small size of transactions, the execution time is increasing with the significant difference. However, when the number of transactions is huge, the execution time is increasing in an ignorable difference. In Fig. 8(c), the curve of total execution time almost has the same linear curve as the one of transform. Because the processing time of *TID* and *P2T* are extremely different, the one of *P2T* is among 6 to 61 and the one of *TID* is among 2 to 18. Obviously the processing time of *P2T* dominate the entire time in the experiment.

5 Conclusions

We in this paper use the RFID readers to detect the user appearance, and with this raw data from readers, we can identify user indoor traversal path. We devise the novel framework, called *ITP*, to mine the top k indoor traversal patterns from user indoor traversal path. While previous works all focus on identifying typical movements of users with certain *ITP* mining in indoor environment, we are the first work to emphasize mining with location uncertainty. To diminish the uncertainty incurred by the RFID-based positioning, we also conduct possible ways according to the general user walking behavior.

In addition, the framework is implemented with empirical studies in order to gain insight into the recall of uncertain *ITP*. The results show that the *ITP* framework has a good stability feature to retrieve high-quality patterns. In the future, we will consolidate the precision of discovered patterns and reduce the false-positive patterns as possible.

Acknowledgment. This paper was supported in part by National Science Council of Taiwan under Contract NSC 101-2221-E-006-246-MY3.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB, pp. 487–499 (1994)
2. Agrawal, R., Srikant, R.: Mining sequential patterns. In: ICDE (1995)
3. Giannotti, F., Nanni, M., Pedreschi, D., Pinelli, F.: Trajectory pattern mining. In: Proceedings of ACM SIGKDD (2007)
4. Han, J., Moraga, C.: The influence of the sigmoid function parameters on the speed of backpropagation learning. In: Sandoval, F., Mira, J. (eds.) IWANN 1995. LNCS, vol. 930, pp. 195–201. Springer, Heidelberg (1995)
5. Jeffery, S.R., Garofalakis, M.N., Franklin, M.J.: Adaptive cleaning for RFID data streams. In: VLDB (2006)
6. Ku, W.-S., Chen, H., Wang, H., Sun, M.-T.: A Bayesian inference-based framework for RFID data cleansing. IEEE Trans. Knowl. Data Eng. **25**, 2177–2191 (2012)
7. Liu, Y., Zhao, Y., Chen, L., Pei, J., Han, J.: Mining frequent trajectory patterns for activity monitoring using radio frequency tag arrays. IEEE Trans. Parallel Distrib. Syst. **23**(11), 2138–2149 (2012)
8. Radaelli, L., Sabonis, D., Lu, H., Jensen, C.S.: Identifying typical movements among indoor objects - concepts and empirical study. In: IEEE 14th International Conference on Mobile Data Management (2013)
9. Ré, C., Letchner, J., Balazinska, M., Suciu, D.: Event queries on correlated probabilistic streams. In: SIGMOD Conference, pp. 715–728 (2008)
10. Sullivan, L.: RFID Implementation Challenges Persist, All This Time Later. InformationWeek, October 2005
11. Tran, T.T.L., Sutton, C., Cocci, R., Nie, Y., Diao, Y., Shenoy, P.J.: Probabilistic inference over RFID streams in mobile environments. In: ICDE, pp. 1096–1107 (2009)
12. Want, R.: The magic of RFID. ACM Queue **2**(7), 40–48 (2004)
13. Welbourne, E., Koscher, K., Soroush, E., Balazinska, M., Borriello, G.: Longitudinal study of a building-scale rfid ecosystem. In: Proceedings of MobiSys (2009)
14. Welle, J., Schulz, D., Bachran, T., Cremers, A.B.: Optimization techniques for laser-based 3D particle filter SLAM. In: ICRA (2010)
15. Yang, B., Lu, H., Jensen, C.S.: Probabilistic threshold k nearest neighbor queries over moving objects in symbolic indoor space (2010)
16. Yang, L., Cao, J., Zhu, W., Tang, S.: A hybrid method for achieving high accuracy and efficiency in object tracking using passive RFID. In: PerCom (2012)
17. Yu, J., Ku, W.-S., Sun, M.-T., Lu, H.: An rfid and particle filter-based indoor spatial query evaluation system (2013)

Myocardial Infarction Classification by Morphological Feature Extraction from Big 12-Lead ECG Data

Julia Tzu-Ya Weng[1](✉), Jyun-Jie Lin[2], Yi-Cheng Chen[3], and Pei-Chann Chang[2]

[1] Department of Computer Science and Engineering,
Yuan Ze University, Taoyuan, Taiwan
julweng@saturn.yzu.edu.tw
[2] Department of Information Management, Yuan Ze University,
Taoyuan, Taiwan
s969203@mail.yzu.edu.tw, iepchang@saturn.yzu.edu.tw
[3] Department of Computer Science and Information Engineering,
Tamkang University, Tamsui, New Taipei City, Taiwan
ycchen@mail.tku.edu.tw

Abstract. Rapid and accurate diagnosis of patients with acute myocardial infarction is vital. The ST segment in Electrocardiography (ECG) represents the change of electric potential during the period from the end of ventricular depolarization to the beginning of repolarization and plays an important role in the detection of myocardial infarction. However, ECG monitoring generates big volumes of data and the underlying complexity must be extracted by a combination of methods. This study combines the advantages of polynomial approximation and principal component analysis. The proposed approach is stable for the 12-lead ECG data collected from the PTB database and achieves an accuracy of 98.07 %.

Keywords: 12-lead ECG · Myocardial infarction · Principal component analysis · Polynomial approximation · Support vector machine

1 Introduction

Myocardial Infarction (MI) is the death of heart due to the sudden blockage of a coronary artery by a blood clot. Blockage of a coronary artery deprives the heart muscle of blood and oxygen, causing injury to the heart muscle. Among the diagnostic tests available to detect heart muscle damage, electrocardiogram (ECG) is one of the most widely used non-invasive diagnostic tools for cardiopulmonary diseases.

ECG monitors the patients' heartbeat and gives accurate and important information about the activities of the atrium and ventricle. A human's normal ECG waveform is shown in Fig. 1. The basic components of an ECG complex are P wave, which represents atrial depolarization, QRS complex, which represents ventricular depolarization and T wave, which corresponds to the period of ventricular repolarization. One normal cardiac cycle starts at the sinus node with the depolarization of the right atrium

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 689–699, 2014.
DOI: 10.1007/978-3-319-13186-3_61

and spreads toward the entire atria in a well-ordered manner. Next, the depolarization impulse reaches the ventricles and the fast contraction produces the QRS complex of the ECG. Finally, ventricular repolarization generates the T-wave complex and the cardiac cycle of one heart beat is terminated [1].

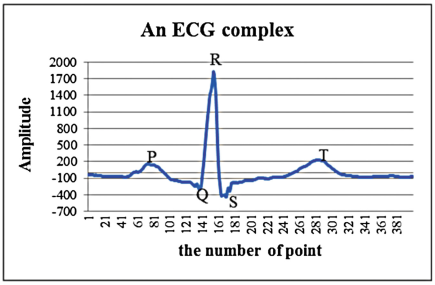


Fig. 1. Basic components of an ECG complex

Clinical 12-lead ECG data are now available in most hospitals and include more detailed information about cardiac disease. The standard 12-lead ECG is composed of six leads: the limb leads, which corresponds to the subject's four extremities, the central terminal, which is the average of the potentials from the limb leads, and six horizontal leads, which are also called chest leads [2]. These leads offer 12 different angles for visualizing the activities of the heart and are named Lead I, II, III, aVL, aVF, aVR, V1, V2, V3, V4, V5 and V6, respectively. Because of the different aspect of recording the polarization and depolarization of a heart-beat cycle, the data volume generated from 12-lead ECG is big and the complexity is high, though more complete information of heart activities can be obtained for cardiac disease classification.

The key in treating ECG complex is using the morphology in time detection [3, 4]. The occlusion of a coronary artery following the rupture of a vulnerable atherosclerotic plaque can represent typical two types of ECG manifestations: ST elevation and ST depression plus T-wave changes [5]. In ECG monitor, ST segment means the change of electric potential during the period which from the end of ventricular depolarization to the origin of repolarization. Hence, ST shape change is a very important parameter for the diagnosis of cardiac disease.

In the early stages of acute MI, the ECG may look normal. Therefore, it is very important to identify a MI from a patient's 12-lead ECG data in the beginning, so that a medical doctor can suggest a patient for expeditious reperfusion therapy and improve prognosis significantly. However, owing to the large volume, great complexity and high dimensionality of 12-lead ECG data, accurately classifying MI and normal data is not a trivial task. Therefore, this study proposes a hybrid approach including polynomial approximation and Principal Component Analysis (PCA) to deal with the challenge in this field. The whole idea is based on studying the effect of feature extraction from ECG data by analyzing the morphological characteristics. In the next section, the related literatures and our analysis workflow will be briefly described.

2 Literature Review

In normal conditions, the ST segment is a horizon line, but in heart diseases, it may show as various waveforms [6]. In MI classification, ST segment change is an important criterion for diagnosis or academic research. Therefore, lots of studies try to use several approaches to extract the features from ST segment in ECG or utilize machine learning techniques to distinguish the difference between normal and MI by using the information in ECG waveforms. We briefly review the related researches below.

2.1 ECG Waveform Analysis in Frequency Domain

Morphological analysis of ECG signals adopts various signal processing strategies over the past two decades. Since ECG complex is a time series data, using Short Time Fourier Transform (STFT) or wavelet can provide a degree of temporal resolution indicating the changes in the frequency spread with time [7]. Wavelet coefficients represent measures of similarity of local shape of the signal with mother and baby wavelets. The multi-resolution properties of WT were effectively utilized for identifying the characteristic points in the ECG waveform and hence for the analysis of PR, RR, ST intervals [8, 9].

2.2 Feature Extraction by Principle Component Analysis

Principle Component Analysis (PCA) is a technique that is generally used for reducing the dimensionality of multivariate datasets [10]. Considering a vector of n random variables x for which the covariance matrix is Σ, the principal components (PCs) can be defined by

$$z = Ax \tag{1}$$

where z is the vector of n PCs and A is the n × n orthogonal matrix with rows that are the eigenvectors of Σ. The eigenvalues of Σ are proportional to the fraction of the total variance accounted for by the corresponding eigenvectors, so that the PCs explaining most of the variance in the original variables can be identified.

PCA has been commonly used to analyze ECG features. The reduced dimensions (features) can be used to represent the beat morphology. For certain periods in ECG, PCA can be also applied to ST-T segment analysis for the detection of myocardial ischemia and abnormalities in ventricular repolarization [11]. PCA in ECG signal processing takes its starting point from the samples of a segment located in some suitable part of the heartbeat. PCA utilizes a representation of the data in a statistical domain rather than a time or frequency domain. In ECG signals, the information can also be separated from the noise or baseline shift by PCA analysis [12].

Some researchers also use PCA to analyze multi-lead ECG [13]. A common way to convert the multi-lead ECG into suitable data is to concatenate it. Ge et al. [14] proposed the research to concatenate the 12-lead in the order: Lead I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, and V6. The dimension of the data will be higher than single-lead, but the information of 12-lead can be solved by PCA at one time.

2.3 Modeling ECG Waveforms by Statistical Models

To solve the classification problem in ECG, lots of researchers use statistical models, e.g. Hidden Markov Models (HMMs) [15]. HMMs are the stochastic models used for representing an underlying stochastic process that is not observable, but can be observed through the sequence of observed symbols. Because of the morphology of ECG signals, HMMs are mostly adopted for classification [16] and segmentation or delineation [17, 18]. HMMs can find the suitable segmentations of a heart-beat by calculating the state transition. Because of its ability to model ECG waveforms, HMMs can also be applied as a feature extractor for artificial intelligence-based classifier [19]. In [19], the log-likelihood calculated from HMMs can be regarded as the feature of a single lead of ECG and this approach can be easily extended to multi-lead diagnosis.

2.4 ST Shape Change Classification by Polynomial Approximation

Polynomial approximation can also be called "curve fitting" or "polynomial fitting". A polynomial is a function that can be written in the form $p(x) = c_0 + c_1x + \ldots + c_nx^n$ for some coefficients $c_0, \ldots, c_n$. If, $c_n \neq 0$ then the polynomial is said to be of order n. A first-order (linear) polynomial is just the equation of a straight line, while a second-order (quadratic) polynomial describes a parabola. The purposes of using polynomial approximation are (1) to model a nonlinear relationship between dependent and independent variables and interest on the shape of the fitted curve and the related coefficients; (2) to approximate a difficult function (e.g. the density or the distribution function).

In the medical field, curve fitting can be applied as a feature extractor of morphological characteristics [20, 21]. For the various shapes in ST segment, some have tried to use polynomial approximation method to extract the features in ECG [22, 23]. The analysis only considers the relative shape change of the ST segment, but this approach can be used to describe the variation of ST shape and provide the important features from the coefficient of polynomials.

2.5 Support Vector Machine

This section briefly describes the basic SVM and non-linear SVM concepts for typical two-class classification problems. Assuming there is a training set with N samples $(X_i, y_i | X_i \in \Re^n, y_i \in \{-1, +1\})$, a hyper-plane can be defined by the following linear function

$$f(X) = \omega^T X + b \tag{2}$$

where w is the weight vector $\{w_1, w_2, \ldots, w_n\}$ and n is the number of attributes (dimensions) and b is a bias. In order to obtain the separating hyper-plane with the largest margin for each training example, the function yields $f(X) \geq 0$ for $y = +1$ and $f(X) < 0$ for $y = -1$. The training set from the two different classes are separated by the hyper-plane $f(X) = 0$ and the SVM classifier is based on the hyper-plane that maximized the separating margin.

The main objective of linear SVM is to maximize the margin and the equation can defined as

$$\begin{aligned} M(w,b) &= \min_{x_i:y_i=-1} d(w,b;x_i) + \min_{x_i:y_i=1} d(w,b;x_i) \\ &= \min_{x_i:y_i=-1} \frac{|\langle w, x_i \rangle|}{\|w\|} + \min_{x_i:y_i=1} \frac{|\langle w, x_i \rangle|}{\|w\|} \\ &= \frac{1}{\|w\|}\left(\min_{x_i:y_i=-1} |\langle w, x_i \rangle + b| + \min_{x_i:y_i=1} |\langle w, x_i \rangle + b| \right) = \frac{2}{\|w\|} \end{aligned} \tag{3}$$

Hence, a minimal problem can be given

$$\begin{cases} \text{minimize } L(w) = \frac{1}{2}\|w\|^2 \\ \text{subject to } y_i(\langle w, x_i \rangle + b) \geq 1 \end{cases} \tag{4}$$

After Lagrangian transformation, we can conclude the dual problem in Eq. (5)

$$\begin{cases} \text{Maximize: } L_D = \sum\limits_{i=1} \alpha_i - \frac{1}{2}\sum\limits_{ij} \alpha_i \alpha_j y_i y_j \left(\langle x_i, x_j \rangle\right) \\ \text{Subject to: } \sum\limits_{i} \alpha_i y_i = 0, \alpha_i \geq 0, \forall i \end{cases} \tag{5}$$

SVMs can be extended to classify nonlinear data through nonlinear kernel mapping function $K(X_i, X_j)$ to replace the original dot operation. The modified function is as follows.

$$\begin{cases} \text{Maximize: } L_D = \sum\limits_{i=1} \alpha_i - \frac{1}{2}\sum\limits_{ij} \alpha_i \alpha_j y_i y_j K\left(x_i, x_j\right) \\ \text{Subject to: } \sum\limits_{i} \alpha_i y_i = 0, \alpha_i \geq 0, \forall i \end{cases} \tag{6}$$

SVMs are one of the kernel-based learning algorithm [22], there exist lots of mapping functions [23] and here the most popular kernel functions are listed.

(1) Linear kernel

$$K(X_i, X_j) = X_i \cdot X_j \tag{7}$$

(2) Polynomial kernel of degree h:

$$K(X_i, X_j) = (X_i \cdot X_j + 1)^h \tag{8}$$

(3) Gaussian radial basis function kernel:

$$K(X_i, X_j) = \exp\left[-\|X_i - X_j\|^2 \Big/ 2\sigma^2\right] \tag{9}$$

(4) Sigmoid kernel:

$$K(X_i, X_j) = \tanh(kX_i \cdot X_j - \delta) \tag{10}$$

As clinical data are linearly inseparable, the nonlinear SVMs are applied and Gaussian RBF is selected as the kernel function in this study. With the kernel mapping function, data from two classes can always be separated by a hyper plane found by using support vectors and margins. In this study, Gaussian RBF kernel has the sigma value of 1 and we use SVM in the Bioinformatics Toolbox of Matlab and randomly selection cross-validation to retrieve the average accuracy.

3 Methodology

In this paper, two approaches for extracting features are compared. Each heartbeat is analyzed in order to increase the accuracy for MI classification. One method is concatenating ST-T segments in 12-lead ECG and then, using PCA to extract the features. The other method is applying PCA on coefficients of polynomial approximation. Figure 2 shows the overall workflow in this study.

3.1 Pre-processing

This study adopts a simple way to decompress the effect of noise and baseline drift. First, a low-pass filter is applied. The threshold of frequency is set as 40. Then each lead is separated into several signals by Empirical Mode Decomposition (EMD) [24], especially suited for nonlinear and non-stationary signals [25]. EMD can be formulated as the following equation:

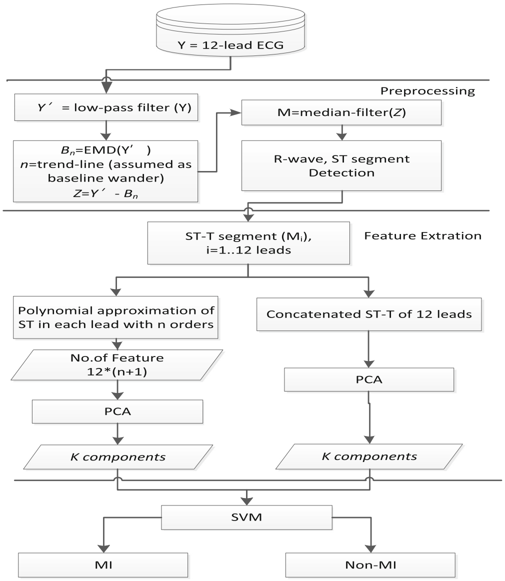
$$X(t) = \sum_{j=1}^{n} IMF_j + r \tag{11}$$

The result of EMD produces n intrinsic mode functions (IMFs) and a residue signal. The residue signal can be regarded as a trend line in the original signal. In this study, the residue signal is regarded as the baseline wander due to the low frequency. After subtracting the assumed baseline wander, a median-filter is used to make the signals more stable and clear.

3.2 Heartbeat Location

QRS-wave location is necessary for ST-T segment identification and heartbeat isolation. To locate QRS wave, ICA is used to process the 12-lead ECG data and the estimated sources are sorted by the kurtosis value calculated from each source. The source with the largest kurtosis value is chosen. Following the approach proposed by [26], heartbeat can be isolated automatically from the complete 12-lead ECG complex. After locating the QRS-wave, the location of R-peak can be defined.

The next is to separate the ST-segment from the whole ECG complex. First, given the R-R interval range between two heartbeats, we assume there is a point J. Second, we calculate the one order difference of the candidate interval and selecting the first minimum value as point J in ECG. As suggested by [6], the threshold for deciding the range of different value is between 0.05 to 0.15. We use the ST segment to diagnosis MI disease, and the end point of ST segment in this section means the peak of T wave. To detect the T peak point in T wave, 12-lead ECG data are superimposed together and calculate the maximum value between point J to the next R peak position. Figure 3 shows the result according to the above steps, and the interval between point J and T wave is picked as the ST segment. Here, we define the ST-T segment as the interval from the beginning of point J, followed by the QRS wave, to the peak of the T-wave.



Y = 12-lead ECG
Preprocessing
Y′ = low-pass filter (Y)
M=median-filter(Z)
B_n=EMD(Y′)
n=trend-line (assumed as baseline wander)
Z=Y′ - B_n
R-wave, ST segment Detection
ST-T segment (M_i), i=1..12 leads
Feature Extration
Polynomial approximation of ST in each lead with n orders
Concatenated ST-T of 12 leads
No.of Feature 12*(n+1)
PCA
PCA
K components
K components
SVM
MI
Non-MI

Fig. 2. The proposed framework

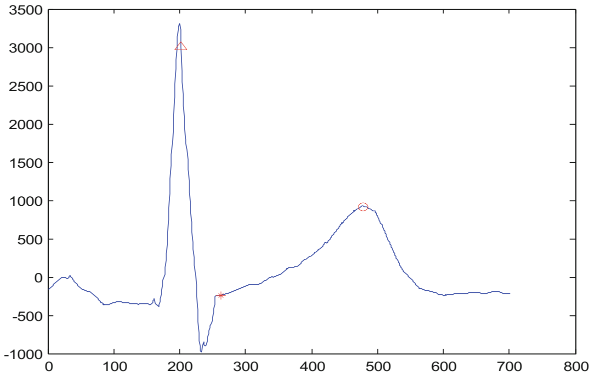


Fig. 3. R peak (Δ), J point (*) and T wave (o) location in an ECG complex

3.3 Feature Extraction by PCA

This study focuses on analyzing whether a single heartbeat belongs to MI in a 12-lead ECG dataset. To combine the information in 12-lead ECG data, the segmented ECG data from the corresponding 12-lead ECG are concatenated in the order of Lead I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, and V6 [21]. PCA is used to reduce and gather significant features.

3.4 Feature Enhanced by PCA

We also adopt another strategy of utilizing PCA to collect coefficients from the polynomial approximation of 12-lead ECG. PCA not only can be used reduce features, but can also be applied to find the most significant features from the original attributes to generate better performance. This study uses polynomial approximation to gather the features of ST segment.

4 Experimental Result

We applied four-fold cross-validation, repeated ten times, to the testing strategy adopted in this research. The performance measurements include accuracy, sensitivity (SE), specificity (SP) and positive predictive (PP). The related equations are listed as below.

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{12}$$

$$SE = \frac{TP}{TP + FN} \tag{13}$$

$$SP = \frac{TN}{TN + FP} \tag{14}$$

$$PP = \frac{TP}{TP + FP} \tag{15}$$

where TP (True Positive) is the number of matched events and FN (False Negative) is the number of events that are not detected by this approach. FP (False Positive) is the number of events detected by this approach. TN (True Negative) indicates the percentage of events identified as truly non-defective, or normal.

The 12-lead ECG data are collected from the PTB-database. There are 549 sets of data, including 148 cases with MI and 301 non-MI cases. The classifier used here is Support Vector Machine and the kernel function selected is RBF kernel with a sigma value of 2. The rule for deciding whether the test case belongs to MI is if more than three-quarters of the number of heartbeats is classified as MI, then the test case is MI; otherwise, the test case will be classified as non-MI case.

Through PCA and polynomial approximation, we try to find the suitable parameters (principle components or PCs) and the appropriate degrees. The number of PCs is initially fixed at 15 and the range of degrees used in polynomial approximation starts from two to ten. We build up a series of testing scenarios, in which the performance of a range of number of PCs is tested with a fixed degree of polynomial approximation. The accuracy increases when the degree of polynomial approximation is four and the number of PCs is larger than seven.

According to our experimental result, we set the PC number at 12 to test a range of degree of polynomial approximation from three to five. Table 1 shows the comparison across different testing models based on their accuracy, sensitivity (SE), specificity (SP) and positive predictivity (PP) measures. These measurements represent the mean values after 30 cross-validations with the corresponding standard deviation. "ST + PCA" indicates the original method of concatenating the ST segments from 12-lead ECG, with features extracted by PCA for SVM. The testing model with the PC number set to 12 and the degree of polynomial approximation fixed at four gives the best overall performance result for MI detection.

Table 1. The performance measurement

		Accuracy	SE	SP	PP
Poly3 + PCA	Mean	96.79 %	**98.74 %**	92.42 %	96.69 %
	std	0.0023	0.0011	0.0077	0.0032
Poly4 + PCA	Mean	**98.07 %**	98.73 %	**96.60 %**	**98.49 %**
	std	0.0029	0.0016	0.0080	0.0035
Poly5 + PCA	Mean	97.96 %	98.71 %	96.26 %	98.34 %
	std	0.0016	0.0013	0.0047	0.0020
ST + PCA	Mean	96.40 %	97.73 %	93.44 %	97.10 %
	std	0.0038	0.0023	0.0115	0.0050

5 Conclusion

PCA and polynomial approximation are considered as two different methods for feature extraction from the ST segment for one heartbeat. We find that these two approaches can be combined to achieve higher performance in MI classification. We further improve the performance of PCA by selecting the proper number of features enhanced by PCA. Since the coefficients of polynomial function can express the variation of ST shape changes, the proposed model indeed increases the performance and reduces the feature space and complexity in a large volume of complex 12-lead ECG data. Through PCA and polynomial approximation, the relationship between ECG and MI disease become more precise.

References

1. Reisne, A.T., Clifford, G.D., Mark, R.G.: The physiological basis of the electrocardiogram. In: Clifford, G.D., Azuaje, F., McSharry, P.E. (eds.) Advanced Methods and Tools for ECG Data Analysis. Artech House Publishing, London (2006)
2. Garcia, T.B., Holtz, N.E.: Introduction to 12-Lead ECG: The Art of Interpretation (2001)
3. de Chazal, P., O'Dwyer, M., Reilly, R.B.: Automatic classification of heartbeats using ECG morphology and heartbeat interval features. IEEE Trans. Biomed. Eng. **51**, 1196–1206 (2004)
4. Reznik, A.G., Ivanov, I.N.: Myocardial morphology in cases of death from acute heart ischemic disease. Arkh. Patol. **69**, 32–35 (2007)
5. Thygesen, K., Alpert, J.S., White, H.D.: Universal definition of myocardial infarction. Circulation **116**, 2634–2653 (2007)
6. Shen, Z., Hu, C., Liao, J., Meng, M.Q.H.: An algorithm of ST segment classification and detection. In: 2010 IEEE International Conference on Automation and Logistics (ICAL'10), pp. 559–564 (2010)
7. Jayachandran, E.S., Joseph, K.P., Acharya, U.R.: Analysis of myocardial infarction using discrete wavelet transform. J. Med. Syst. **34**, 985–992 (2010)
8. Addison, P.S.: Wavelet transforms and the ECG: A review. Physiol. Meas. **26**, R155–R199 (2005)
9. Ghaffari, A., Homaeinezhad, M.R., Akraminia, M., Atarod, M., Daevaeiha, M.: A robust wavelet-based multi-lead electrocardiogram delineation algorithm. Med. Eng. Phys. **31**, 1219–1227 (2009)
10. Jolliffe, I.T.: Principal Component Analysis. Springer, New York (1986)
11. Murugan, S., Radhakrishnan, S.: Automated ischemic beat classification using Genetic Algorithm based Principal Component Analysis. Int. J. Healthc. Technol. Manage. **11**, 151–162 (2010)
12. Chawla, M.P.S.: PCA and ICA processing methods for removal of artifacts and noise in electrocardiograms: A survey and comparison. Appl. Soft Comput. **11**, 2216–2226 (2010)
13. Castells, F., Laguna, P., Sörnmo, L., Bollmann, A., Roig, J.M.: Principal component analysis in ECG signal processing. EURASIP J. Adv. Sig. Process. **2007**, 1–21 (2007). Article ID: 074580
14. Ge, D., Sun, L., Zhou, J., Shao, Y.: Discrimination of myocardial infarction stages by subjective feature extraction. Comput. Methods Programs Biomed. **95**, 270–279 (2009)

15. Chauhan, S., Wang, P., Sing, L.C., Anantharaman, V.: A computer-aided MFCC-based HMM system for automatic auscultation. Comput. Biol. Med. **38**, 221–233 (2008)
16. Andreao, R.V., Dorizzi, B., Boudy, J., Mota, J.C.M.: ST-segment analysis using hidden Markov model beat segmentation: Application to ischemia detection. Comput. Cardiol. **31**, 381–384 (2004)
17. Graja, S., Boucher, J.M.: Hidden Markov tree model applied to ECG delineation. IEEE Trans. Instrum. Meas. **54**, 2163–2168 (2005)
18. Andreao, R.V., Dorizzi, B., Boudy, J., Mota, J.C.M.: ST-segment analysis using hidden Markov model beat segmentation: Application to ischemia detection. Comput. Cardiol. **31**, 381–384 (2004)
19. Chang, P.C., Hsieh, J.C., Lin, J.J., Chou, Y.H., Liu, C.H.: A Hybrid System with Hidden Markov Models and Gaussian Mixture Models for Myocardial Infarction Classification with 12-Lead ECGs. In: Proceedings of the 2009 11th IEEE International Conference on High Performance Computing and Communications (HPCC '09), pp. 110–116 (2009)
20. Georgiou, H., Mavroforakis, M., Dimitropoulos, N., Cavouras, D., Theodoridis, S.: Multi-scaled morphological features for the characterization of mammographic masses using statistical classification schemes. Artif. Intell. Med. **41**, 39–55 (2007)
21. Li, B., Meng, M.Q.H.: Computer-based detection of bleeding and ulcer in wireless capsule endoscopy images by chromaticity moments. Comput. Biol. Med. **39**, 141–147 (2009)
22. Jeong, G.Y., Yu, K.H., Kim, N.G.: A polynomial approximation approach for analyzing ST shape change. In: the 27th Annual International Conference of the IEEE Engineering in Medicine and Biology (EMBS'05), Vol. 7, pp. 4034–4037 (2005)
23. Jeong, G.Y., Yu, K.H., Yoon, M.J., Inooka, E.: ST shape classification in ECG by constructing reference ST set. Med. Eng. Phys. **32**, 1025–1031 (2010)
24. Blanco-Velasco, M., Weng, B., Barner, K.E.: ECG signal denoising and baseline wander correction based on the empirical mode decomposition. Comput. Biol. Med. **38**, 1–13 (2008)
25. Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.C., Tung, C.C., Liu, H.H.: The empirical mode decomposition and the Hubert spectrum for nonlinear and non-stationary time series analysis. Proc. Royal Soc. A: Math. Phys. Eng. Sci. **454**, 903–995 (1998)
26. Chawla, M.P.S., Verma, H.K., Kumar, V.: A new statistical PCA-ICA algorithm for location of R-peaks in ECG. Int. J. Cardiol. **129**, 146–148 (2008)

Automatic Restaurant Information and Keyword Extraction by Mining Blog Data for Chinese Restaurant Search

Chien-Li Chou[1(✉)], Min-Ho Tsai[1], Chien-Ho Chao[1],
Hsiao-Jung Lin[1], Hua-Tsung Chen[2], Suh-Yin Lee[1],
and Chien-Peng Ho[3]

[1] Department of Computer Science, National Chiao Tung University,
1001 Dahsueh Road, Hsinchu 30010, Taiwan
{fallwind, sylee}@cs.nctu.edu.tw,
{shjk4210, pass518224, smilecatxiii}@gmail.com

[2] Information and Communications Technology Lab,
National Chiao Tung University, 1001 Dahsueh Road,
Hsinchu 30010, Taiwan
huatsung@cs.nctu.edu.tw

[3] ICL/Industrial Technology Research Institute, 195 Chung Hsing Road,
Section 4, Chutung, Hsinchu, Taiwan
cpho@itri.org.tw

Abstract. Restaurant search and recommendation system is a very popular service in many countries. In those systems, most of the restaurant information such as restaurant name, address, phone number, and introduction are collected manually. In this paper, we propose a restaurant information extraction method which can automatically extract restaurant information from online reviews of restaurants in blogs. In addition, by calculating TFIDFs of words in blog posts, the hot keywords can be discovered and ranked. For restaurant search, users are allowed to search by keywords, areas, and/or extracted hot keywords. The experimental results show that the proposed method can achieve over 90 % average accuracy of hot keyword extraction and about 95 % mean average precision for restaurant search. In user study, the fact that the proposed system is more useful than Google search in restaurant search is presented.

Keywords: Information retrieval · Opinion mining · TFIDF · Food and restaurants · Restaurant search

1 Introduction

With the rapid growth and affordable cost of Internet bandwidth, more and more web contents are generated by not only business content providers but also customers and users. Nowadays, blogs, the web pages for users to post their words, are widely spread and used. People post their moods, thoughts, and comments for something such as what they buy, where they go, and what they eat. According to the statistics from MBAonline.com in 2012, two million blog posts were written in one day. Such a huge

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 700–711, 2014.
DOI: 10.1007/978-3-319-13186-3_62

number of data make the search results noisy and redundant. Therefore, mining useful information in such big data becomes a vital issue.

Many blog posts are written for recording the dining. Users write down their comments for the food and environment in restaurants they went to. This kind of blog posts is not only a record of life but also useful information for other people. For example, when people want to have dinner in an unfamiliar city, they usually search the reviews about restaurants there on the Internet by keywords. However, users have to spend much time to find and read the unorganized reviews. People usually want to collect more reviews about the desired restaurant to confirm if it is really good. Thus, an automatic restaurant information and keyword extraction system can reduce the time spent on collecting the similar reviews of restaurants and can extract correct restaurant information such as address and phone number. Some websites such as Yelp [1], Tabelog [2], and HOT PEPPER [3] provides restaurant search services to users for America, English, and Japanese restaurants. For Chinese restaurants, there are few restaurant search sites as good as the above sites. Therefore, we focus on restaurant information and keyword extraction for Chinese restaurants by mining contents of blog posts.

In this paper, we propose a method for automatic restaurant information and keyword extraction based on the techniques of information retrieval and pattern mining. Using the extracted information and keywords, we can develop a crowd sourcing restaurant search system to view the real comments from other people instead of business campaigns.

The remainder of this paper is organized as follows. The related literatures are reviewed in Sect. 2. The proposed method for restaurant information and keyword extraction is described in detail in Sect. 3. In Sect. 4, comprehensive experiments including quantitative and qualitative evaluations are conducted and the experimental results are presented. Finally, we conclude this work in Sect. 5.

2 Related Work

To extract the restaurant information, such as restaurant name, address, and phone number, from unstructured text content of blog posts, called “named entity recognition [4]”, many studies for English websites were well conducted [5, 6]. However, it is hard to recognize the named entities in Chinese since the quality of Chinese word segmentation technique is insufficient to segment the correct phrase of named entities.

Many researchers focused on opinion mining from the online reviews. Hu et al. [7] created rules based on the number of frequencies from user reviews to extract the product related characteristics, and then classified the reviews into positive ones or negative ones by the extracted product characteristics. Jindal et al. [8] analyzed the components of the comparative sentences discovered from online reviews to extract the comparative targets and characteristics. The comparative sentences are categorized into four types. For each type appropriate rules were generated and applied to extract the comparative advantage target. Gu et al. [9] mined popular menu items of restaurants from web reviews by analyzing the post frequencies. Kato et al. [10] extracted the onomatopoeia from the online reviews by calculating the TFIDF of words and used the onomatopoeia to search the desired restaurants of users.

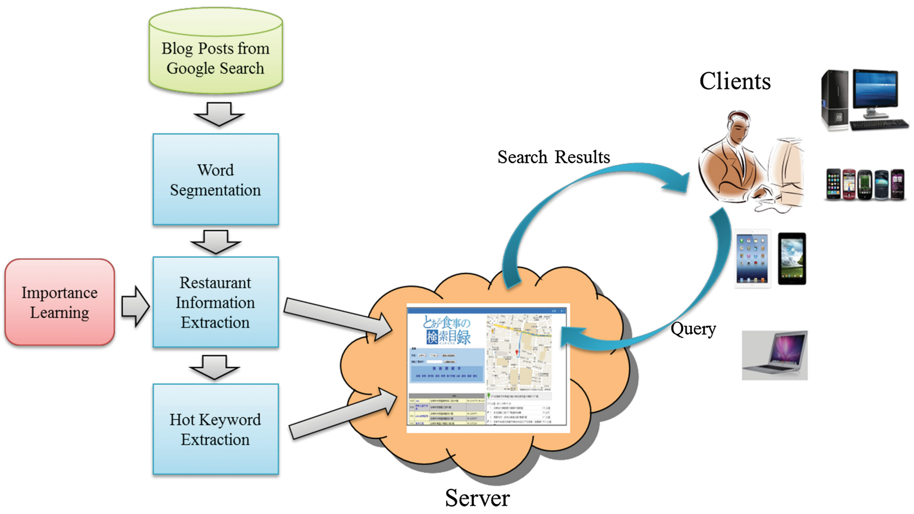


Fig. 1. The system framework

As an application, restaurant recommendation is an interesting topic for researchers. Yu et al. [11] developed a context-aware travel planning system which can recommend where to live, where to go, and where to dine. Gupta et al. [12] proposed a personalized location based restaurant recommendation system. The user preferences and location were taken into consideration to recommend the restaurants. Chu et al. [13] also developed a context-aware Chinese restaurant recommendation system. Kitayama et al. [14] constructed a restaurant information retrieval system by learning the relations among search properties based on the operational context. Association rule mining is applied to extract the relations among search properties.

3 Proposed Method

The proposed system can be divided in to two stages: (1) offline restaurant information and keyword extraction stage and (2) online restaurant search stage. The system framework is shown in Fig. 1. In the offline stage, the restaurant information such as store name, telephone number, and address is extracted, and the keywords of restaurants are then computed by analyzing contents of blog posts. In the online search and recommendation stage, users can type the keyword to search the desired restaurants, and the proposed system will return the related restaurants to users. Furthermore, the system can recommend the restaurant indirectly related to the keyword. The details of the two stages are described in the following.

3.1 Offline Information and Keyword Extraction Stage

As shown in Fig. 2, the offline stage consists of four steps is described below.

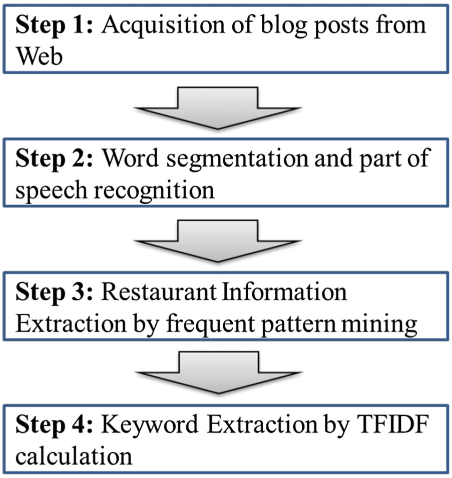

Step 1: Acquisition of blog posts from Web
Step 2: Word segmentation and part of speech recognition
Step 3: Restaurant Information Extraction by frequent pattern mining
Step 4: Keyword Extraction by TFIDF calculation

Fig. 2. The flow chart of offline information and keyword extraction

Step 1. Acquisition of blog posts

In this step, we collect the blog posts related to restaurants from Google search engine. We use one keyword "dining record" (in Chinese) and another keyword for area such as Taipei, etc. to search the blog posts containing reviews of restaurants. The keyword for area can reduce the noise in search results since it narrow the search space. Every returned result page $\boldsymbol{P_i}$ is then processed to extract the restaurant information.

Step 2. Word segmentation and part of speech (POS) recognition

For each page returned by Google, we parse the title T_i and main body B_i of the result page P_i. The parsed texts are then analyzed by CKIP Chinese word segmentation system [15]. CKIP can segment Chinese sentences into phrases or words and recognize their POS. We only keep nouns for the following steps since names of restaurants and keywords are usually nouns. Therefore, after the segmentation and filtering, a set of nouns $N^{T_i} = \{N_1^{T_i}, N_2^{T_i}, \ldots, N_X^{T_i}\}$ and another set $N^{B_i} = \{N_1^{B_i}, N_2^{B_i}, \ldots, N_Y^{B_i}\}$ are obtained from T_i and B_i, respectively.

Step 3. Restaurant Information Extraction

To extract restaurant information, first, we extract names of restaurants. According to observations, the title of a blog post for dining record usually consists of the name of the restaurant introduced in the post. That is, if a noun appears in N^{T_i} and N^{B_i} at the same time, it is most likely part of the name of the restaurant. If there are more than one nouns consecutively appearing in the same order in both the sets, the nouns are concatenated to form an extended noun as a candidate. For example, assume that the title of a blog post is "Noodle Store: the good place for lunch", and the main body is "Looking for lunch? Come to Noodle Store." After segmentation and filtering, the title is segmented into nouns "noodle", "store", "place", and "lunch", and the main body becomes "lunch", "noodle", and "store". Since the words "noodle" and "store" appear consecutively and in the same order in both sets, we can concatenate the two words to "noodle store." The word "lunch" appears in both sets, so it also becomes a candidate. "Noodle store" and "lunch" cannot be combined since the orders of these two words are not the same. To select one of the candidates to be the name of the restaurant, we define the name score S_{name} as

$$S_{name}(C) = \alpha \times Importance(C) + (1 - \alpha) \times Freq(C) \times Length(C), \tag{1}$$

where C is a candidate, α is the weight, $Freq(C)$ is the candidate appearing frequency in the post, $Length(C)$ is the number of words in C, and $Importance(C)$ is the importance of C. To define the Importance of a candidate, we randomly collect 300 names of restaurants as training data to train the importance of words. A name of restaurant is regarded as a transaction, and a word is regarded as an item. Frequent pattern mining [16, 17] is then applied to find the common words for names of restaurants. For a candidate C, words in C is regarded as items. Then we generate a set I_C containing all possible itemsets $\{I_1, I_2, \ldots\}$ for C. The importance of candidate C can be defined as

$$Importance(C) = \sum\nolimits_{I_i \in I_C} (Length(I_i) \times Support(I_i)), \text{ if } I_i \text{ is in } FPS, \tag{2}$$

where FPS is the frequent pattern set, $Length(I_i)$ is the number of items of I_i, and $Support(I_i)$ is the support of the frequent pattern corresponding to I_i. The candidate with the highest name score is selected to be the name of the restaurant in the post. After selecting the name of the restaurant, the other restaurant information is then extracted by searching the text nearby the name in main body. Usually, telephone numbers are in some specific formats, such as 0x-xxx-xxxx, 09xx-xxx-xxx, etc. Using this constraint, we can easily extract the phone number. Simultaneously, the address is extracted by searching the area names downloaded from the website of post office.

To validate the correctness of the extracted name, address and phone number of a post, we use the extracted name and the area as the keyword to search by Google. We select the top K results and repeat step 1 to 3. For each result, we can obtain a set of name, address, and phone number. The final restaurant information is decided by major voting scheme. If no information is voted by more than one page, we may select the wrong name of the restaurant. Hence, we select the next name candidate as the name, and repeat the above procedure until a correct name is found or no candidate can be processed.

Step 4. Keyword Extraction

Extracting keywords from the text of a blog post is an important task for restaurant search and indexing. For keyword extraction, we calculate the TFIDF value of every word. TF and IDF for a noun N_j in the post P_i are defined as

$$TF_{N_j}^{P_i} = \frac{n_{N_j}^{P_i}}{|N^{T_i}| + |N^{B_i}|} \tag{3}$$

$$IDF_{N_j} = \log \frac{|P|}{\left|\left\{i : N_j \in P_i\right\}\right|}, \tag{4}$$

where $n_{N_j}^{P_i}$ is the number of N_j appearing in P_i. And the TFIDF of a keyword N_j for a restaurant R is defined as

$$TFIDF_{N_j}^{R} = \sum\nolimits_{P_i \in R} TF_{N_j}^{P_i} \times IDF_{N_j}. \tag{5}$$

The nouns with top 20 high TFIDF are selected as the keywords of the restaurant. For each area, we aggregate all the keywords of restaurants in the area, and accumulate the TFIDF values of the same keywords of different restaurants. The top 10 keywords are selected to be the area keywords, which can be recommended to users as hot keywords.

However, the long phrases may be segmented in the step of word segmentation, which causes that a long phrase is hard to be a keyword. Therefore, we propose a keyword expansion method to recover the long keywords. For each keyword of a restaurant, we search the main body to find the position of the keyword. If the previous and next words of the keyword in main body are nouns, we concatenate those words together to form an expanded keyword. For example, “noodle” is a keyword of the restaurant. “Seafood noodle” appears in the main body but the word “seafood” is not a keyword. We can concatenate “seafood” and “noodle” to form an expanded keyword since “noodle” is a keyword and the previous word of “noodle”, “seafood,” is a noun. That is, with keyword expansion, when a user search by long keyword, the system can still return the correct results.

3.2 Online Search Stage

Search by Keyword
Users can type their desired keywords or choose one of the hot keywords extracted from blog posts, as the interface shown in Fig. 3. The restaurants contain the keyword(s) are retrieved and ranked by the TFIDF of the keyword(s) in the restaurant. Users can view the related paragraphs of the corresponding blog posts in the system for judging if they go to the restaurant. As elaborated in Fig. 4, this mechanism emphasizes the texts related to the keyword and provides a quick review of the blog posts for the restaurant to users.

Search by Location
Users can search the nearby restaurants if the system obtains the location information from the device or inputted by users. Google Map is used to calculate the distances between the location of users and the restaurant addresses. Google navigation can plan the routes among multiple destinations. Search by location and search by keyword can work together.

4 Experimental Evaluation

4.1 Prototype System

We develop a prototype system to evaluate the proposed method for restaurant search. Figure 4 shows the interface of the proposed interface consisting of area search

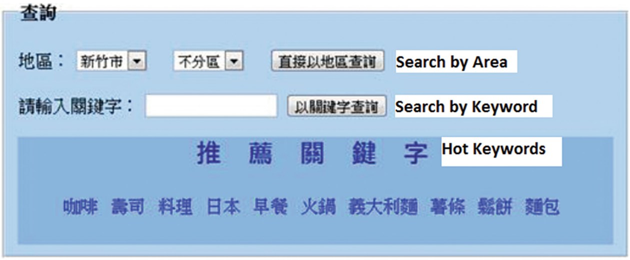
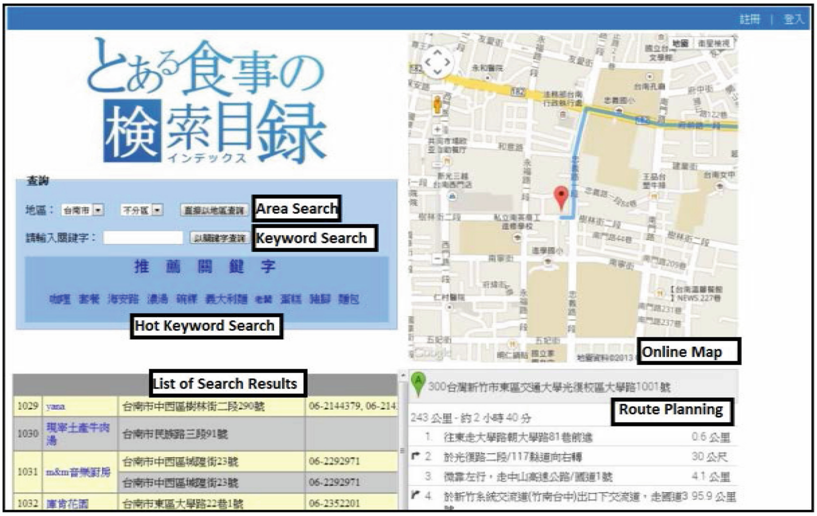


Fig. 3. The search interface of the proposed system



1029	yana	台南市中西區樹林街二段290號	06-2144379, 06-214
1030	現宰土產牛肉湯	台南市民族路三段91號	
1031	m&m音樂廚房	台南市中西區城隍街23號	06-2292971
		台南市中西區城隍街23號	06-2292971
1032	庫肯花園	台南市東區大學路22巷1號	06-2352201

300台灣新竹市東區交通大學光復校區大學路1001號

243 公里 - 約 2 小時 40 分

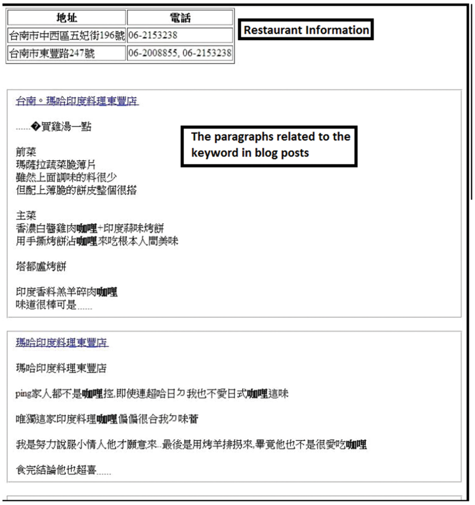
1. 往東走大學路朝大學路81巷前進 0.6 公里
2. 於光復路二段/117縣道向右轉 30 公尺
3. 微靠左行，走中山高速公路/國道1號 4.1 公里
4. 於新竹系統交流道(竹南台中)出口下交流道，走國道3號 95.9 公里

Fig. 4. The prototype System

component, keyword search component, hot keyword search component, list of search results component, online map component, route planning component. The online map and route planning component applies Google Map API to acquire the user location and to plan the route to restaurants. By clicking a restaurant name listed in the search result, the page will show all the blog posts related to the restaurant, as shown in Fig. 5.

4.2 Experimental Setting

We collect blog posts for restaurants in 19 areas in Taiwan as listed in Table 1. The keyword for Google search is set to "dining record AREA" (in Chinese), where AREA is the name of an area. At least 50 restaurants for each area and at least 4 blog posts for each restaurant are collected. Totally 1099 restaurants and 5483 blog posts are used to conduct the experiments. For better evaluation of the proposed method, we conduct both quantitative and qualitative experiments.



地址	電話
台南市中西區五妃街196號	06-2153238
台南市東豐路247號	06-2008855, 06-2153238

Restaurant Information

台南。瑪哈印度料理東豐店

……�買雞湯一點

前菜
瑪薩拉蔬菜脆薄片
雖然上面調味的料很少
但配上薄脆的餅皮整個很搭

主菜
香濃白醬雞肉**咖哩**+印度蒜味烤餅
用手撕烤餅沾**咖哩**來吃根本人間美味

塔都爐烤餅

印度香料羔羊碎肉**咖哩**
味道很棒可是……

The paragraphs related to the keyword in blog posts

瑪哈印度料理東豐店

瑪哈印度料理東豐店

ping家人都不是**咖哩**控,即使連超哈日ㄉ我也不愛日式**咖哩**這味

唯獨這家印度料理**咖哩**偏偏很合我ㄉ味蕾

我是努力說服小情人他才願意來..最後是用烤羊排拐來,畢竟他也不是很愛吃**咖哩**

食完結論他也超喜……

Fig. 5. The quick review of the blog posts for the selected restaurant

Table 1. List of Taiwan areas used to search the blog posts

List of the areas used in the proposed method		
台北市 (Taipei city)	彰化縣 (Changhua county)	屏東縣 (Pingtung county)
新北市 (New Taipei city)	雲林縣 (Yunlin county)	基隆市 (Keelung city)
桃園縣 (Taoyuan county)	南投縣 (Nantou county)	宜蘭市 (Yilan county)
新竹縣 (Hsinchu county)	嘉義縣 (Chiayi county)	花蓮縣 (Hualien county)
新竹市 (Hsinchu city)	嘉義市 (Chiayi city)	台栗縣 (Taitung county)
苗栗縣 (Miaoli county)	台南市 (Tainan city)	
台中市 (Taichung city)	高雄市 (Kaohsiung city)	

Quantitative Experiments

For quantitative experiments, we apply three measurements for evaluating the accuracy of hot keyword extraction, the average precision (AP) of search results, and the mean average precision (MAP) of search results. Within an area, the accuracy of hot keyword extraction is defined as

$$\mathrm{Acc} = \frac{\#\, representative\ hot\ keywords}{\#\, hot\ keywords\ extracted}. \tag{6}$$

Whether a hot keyword is representative or not is decided by users.

Given a set of hot keywords $H = \{H_1, H_2, \ldots, H_Z\}$, the AP within an area is defined as

$$\mathrm{AP} = \frac{\sum_i Precision(H_i)}{|H|}, \tag{7}$$

where *Precision* (H_i) is the precision of the search results of H_i. And the MAP for the proposed system is defined as

$$\mathrm{MAP} = \sum\nolimits_i AP_i / \#\, areas. \tag{8}$$

Qualitative Experiments

We invite 12 university students to rate the user experiences of our proposed system. Each participant performs 19 search tasks (One search task for one area) and rates score 1 ~ 5 on the following options.

- Convenience (1 is not convenient for users, and 5 is convenient.)
- Practicability (1 is less practicability, and 5 is more practicability.)
- Smoothness on use (1 is hard on use, and 5 is smooth on use.)
- Is it better than Google search for restaurant search? (1 means Google is much better than the proposed system, and 5 means the proposed system is much better than Google.)

4.3 Experimental Results

Figure 6 shows the accuracy of hot keyword extraction. Most of the accuracies of hot keyword extraction are greater than 90 %. That is, the extracted hot keywords are representative enough for users. However, in Taoyuan County, the accuracy of hot keyword extraction is only 70 %. It means that in all extracted 10 hot keywords, three of them are uninformative to be search keywords. The reason is that some of the blog posts collected in the area of Taoyuan County are written by the same author. The author often uses nicknames in his posts, and the nicknames are usually extracted as keywords since the TF and IDF of the words are high. The average accuracy of hot keyword extraction is 91.58 %, which provides sufficient information for users.

The average precisions are illustrated in Fig. 7. Same as the accuracy of hot keyword extraction, most of APs are high enough to provide correct search results to users. In some specific areas, about only 80 % APs are achieved, and we observe that some conditions of the areas. First, the areas with low APs are less famous in restaurants so that few people write the blog posts for restaurants in those areas. Second, a famous night market is in the area. Blog posts written for the night market make the restaurant information noisy because there are many vendors in the night market. The author introduces multiple vendors, and the information of different vendors is mixed and hard to separate. As a result, a blog post focuses on multiple restaurants makes the search results noisy. The MAP for restaurant search is 94.85 %, which is high enough for users to obtain desired restaurant information.

4.4 User Study

In the qualitative experiments, the average convenience score rated by 12 participants is 4.25, the average practicability score is 3.75, and the average smoothness score is 4.0. The result shows that the participants put a premium on the proposed system in

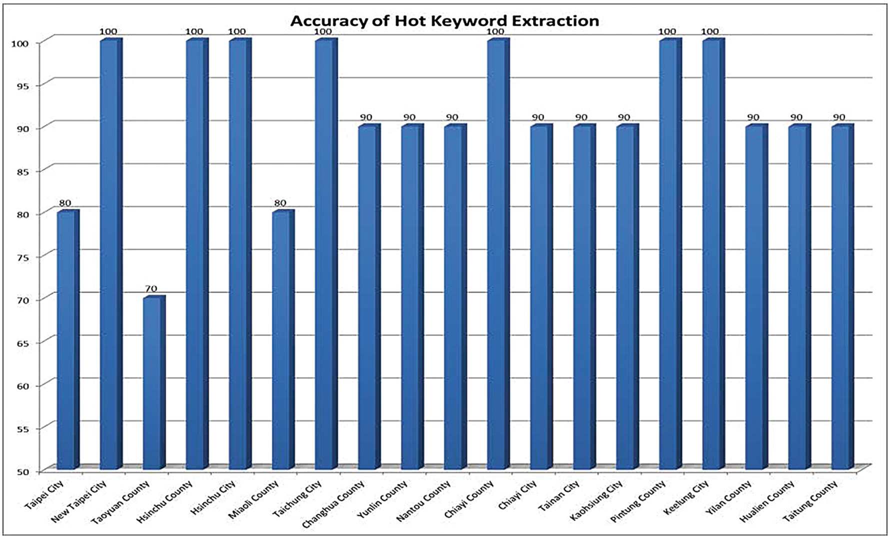


Fig. 6. The accuracy of hot keyword extraction

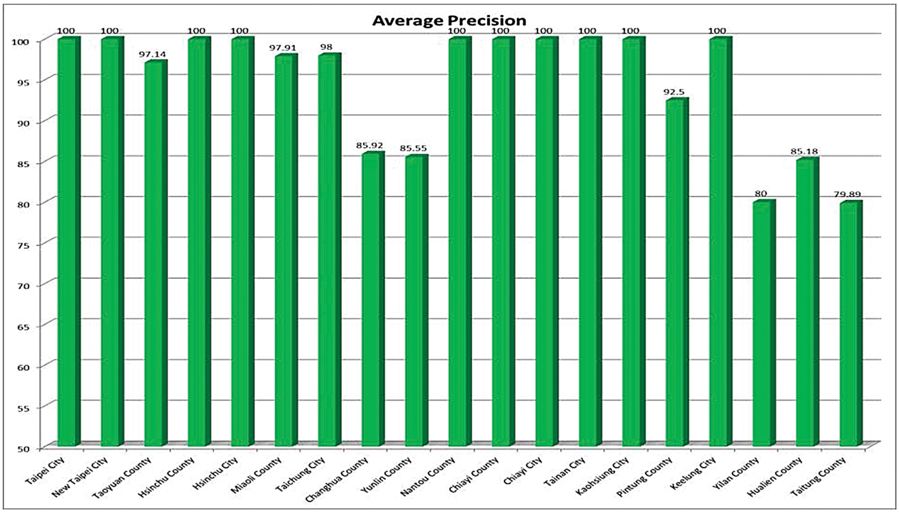


Fig. 7. The average precision of areas

convenience and smoothness. That is, the proposed system is well designed for good user experience. Because the number of restaurants is not sufficient to make every participant satisfied, the practicability score is lower than the other two scores. In the last question “Is it better than Google search for restaurant search?” of this user study, the average score is 4.75. Most of all participants give 5 points for this question. It depicts that the proposed system is really useful for users who want to search a restaurant.

5 Conclusion

We proposed an automatic restaurant information and keyword extraction system by mining the blog posts data for restaurant search. From the large number of blog posts, the restaurant information such as restaurant name, address, and phone number can be automatically extracted and validated by pattern mining techniques. By using the TFIDF approach, the representative words are extracted to be hot keywords for users' references. The experiment results show that the average accuracy of hot keyword extraction is over 90 % and the MAP of restaurant search is about 95 %. In user study, we observe that the extracted hot keywords and the restaurant information are more compact and practicable than the information searched from Google search.

References

1. Yelp. http://www.yelp.com
2. Tabelog. http://tabelog.com/
3. HOT PEPPER. http://www.hotpepper.jp/
4. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Invest. **30**(1), 3–26 (2007)
5. Alfonseca, E., Manandhar, S.: An unsupervised method for general named entity recognition and automated concept discovery. In: 1st International Conference on General WordNet, pp. 1–9 (2002)
6. Satoshi, S., Nobata, C.: Definition, dictionaries and tagger for extended named entity hierarchy. In: 4th International Conference on Language Resources and Evaluation, pp. 1977–1980 (2004)
7. Hu, M. Liu, B.: Mining and summarizing customer reviews. In: 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177 (2004)
8. Jindal, N., Liu, B.: Mining comparative sentences and relations. In: 21th National Conference on Artificial Intelligence, vol. 2, pp. 1331-1336 (2006)
9. Gu, Y.H., Yoo, S.J.: Mining popular menu items of a restaurant from web reviews. In: Gong, Z., Luo, X., Chen, J., Lei, J., Wang, F.L. (eds.) WISM 2011, Part II. LNCS, vol. 6988, pp. 242–250. Springer, Heidelberg (2011)
10. Kato, A., Fukazawa, Y., Sato, T., Mori, T.: Extraction of onomatopoeia used for foods from food reviews and its application to restaurant search. In: 21st International Conference Companion on World Wide Web, pp. 719–728 (2012)
11. Yu, C.C., Chang, H.P.: Towards Context-Aware Recommendation for Personalized Mobile Travel Planning. In: Vinh, P.C., Hung, N.M., Tung, N.T., Suzuki, J. (eds.) ICCASA 2012. LNCS, vol. 109, pp. 121–130. Springer, Heidelberg (2012)
12. Gupta, A., Singh, K.: Location based personalized restaurant recommendation system for mobile environments. In: International Conference on Advances in Computing, Communications and Informatics, pp. 507–511 (2013)
13. Chu, C.H., Wu, S.H.: A Chinese restaurant recommendation system based on mobile context-aware services. In: 14th International Conference on Mobile Data Management, vol. 2, pp. 116–118 (2013)

14. Kitayama, D., Matsuo, J., Sumiya, K.: Extracting relations among search properties based on the operational context of geographical information retrieval systems. In: Hong, B., Meng, X., Chen, L., Winiwarter, W., Song, W. (eds.) DASFAA Workshops 2013. LNCS, vol. 7827, pp. 179–192. Springer, Heidelberg (2013)
15. CKIP Chinese Word Segmentation System. http://ckipsvr.iis.sinica.edu.tw/
16. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM SIGMOD International Conference on Management of Data, pp. 1–12 (2000)
17. Li, H., Wang, Y., Zhang, D., Zhang, M., Chang, E.Y.: PFP: parallel FP-growth for query recommendation. In: ACM Conference on Recommender Systems, pp. 107–114 (2008)

The Application of Association Rules in Clinical Disease: The Relationship Between Recovery After Operation of Endovascular Aneurysm Repairing and Chronic

Lin Hui[1](✉), Chun-Che Shih[2], Huan-Chao Keh[3], Po-Yuan Yu[3], Yuan-Cheng Cheng[3], and Nan-Ching Huang[3]

[1] Department of Innovative Information and Technology,
Tamkang University, New Taipei, Taiwan
amar0627@gmail.com

[2] Institute of Clinical Medicine, National Yang-Ming University, Taipei, Taiwan
ccshih@vghtpe.gov.tw

[3] Department of Computer Science, Tamkang University, New Taipei, Taiwan
{Keh,144920,132310}@mail.tku.edu.tw,
fishyu750502@hotmail.com

Abstract. This research is carried out in order to find out the effects of chronic which posed on the recovery after operation of endovascular aneurysm repairing in the method of associational data mining by analyzing the number of days which the patient spending in hospital who had finished the operation of endovascular aneurysm repairing; by the result of this research, we will find out the chronic that can seriously affect the recovery and make it a reference for medical personnel like clinicians so that they can target the chronic preventing the potential harm happening ahead of time. It can also improve the recovery after operation and achieve the goal of reducing days in hospital, as well as the waste of medical resource.

Keywords: Aortic aneurysm · Endovascular aneurysm repair · Chronic disease · Association rules

1 Introduction

According to the Ministry of Health and Welfare [4], People have been dead of Endovascular aneurysm reached the number of 714 during 2012, that means about 2 people died each day because of Endovascular aneurysm; blood vessel's deterioration is one of the significant reasons. With the age increasing, blood vessel's deterioration will result in Endovascular aneurysm. During the recent years, percentage of the elderly in Taiwan has been growing and forms a new great threat of Endovascular aneurysm that Taiwan society will have to face in the future.

There are two main methods to cure the Endovascular aneurysm, one is traditional operation, and the other is EVAR (Endovascular aneurysm repair). The traditional is to operate a thoracotomy surgery or Laparotomy directly and leave a larger wound on

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 712–721, 2014.
DOI: 10.1007/978-3-319-13186-3_63

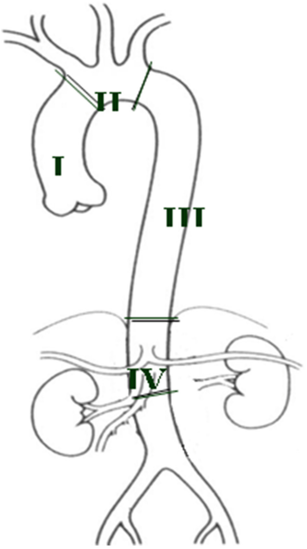
chest compared to minimally invasive surgery. That will cost the patient over one month to stay in hospital. However, EVAR only needs a 3-cm wound which is wide enough to do the operation. Because of this, patients can go home after a week living in hospital.

In this research, we take the association rules to do analysis on patients and find out the key chronic which can pose harm on the after-operation recovery, make it a reference for clinicians so that they can target the chronic preventing the potential harm happening ahead of time, improve the recovery after operation and achieve the goal of reducing days in hospital, as well as the waste of medical resource.

2 Related Work

2.1 Aorta Aneurysm

The formation of aorta aneurysm is mainly caused by the calcification and degeneration of arterial wall as well as some other factors. These factors result in the lack of flexibility on arterial wall. When this appears on aorta, the aorta aneurysm comes out. The worse is that aorta may burst and cause massive internal bleeding even death till the diameter reach to 6 cm. The classification of aorta aneurysm is mainly according to the place where it happens. The major types of aorta aneurysm can be aneurysm of thoracic aorta which happens on thoracic aorta known as aorta ascendens, arcus aortae and aorta descendens; and abdominal aneurysm which happens on aorta ventralis. However, there's another one called dissecting aortic aneurysm when the lining of aorta wall burst causing blood flow into aorta wall forming cavum septi pellucidi aorta aneurysm (Fig. 1).



I	Aorta ascendens
II	Aortic arch
III	Aorta descendens
IV	Abdomial aorta

Fig. 1. Clasification of the place in aorta [7]

2.2 Endovascular Aneurysm Repair

Endovascular aneurysm repair is an original minimally invasive surgery which was attached importance by Parodi J.C. [6] and other expects since 1991 in Endovascular aneurysm curing application. Differing from the traditional thoracotomy surgery or Laparotomy, Endovascular aneurysm repair only needs a 3-cm wound which is wide enough to do the operation. During the operation, a stent catheter will be put at the narrow place in aneurysm directly guided by a line via aneurysm. Then, stent catheter will be pulled back slowly releasing a stent graft which let the blood flow smoothly through the aneurysm reaching the goal of repairing aneurysm inside aneurysm. For the 3-cm wound is the only needed, this operation can greatly share the burden of wound which brought form itself. This advantage also leads to another result that patients will take less risk in operation and the mortality rate is much more lower than the traditional one [2, 5].

The stents set in the operation of endovascular aneurysm repair can be divided into two types, one is self-expandable stent, the other is balloon-expandable stent. Figure 2 shows the placement process of self-expandable stent. As we can see, firstly lead the guide line to the affected part along with the aorta, then place the self-expandable stent catheter to the part through the guiding of the line set before, now it's the turn of pulling back the catheter releasing the self-expandable stent graft. During the whole

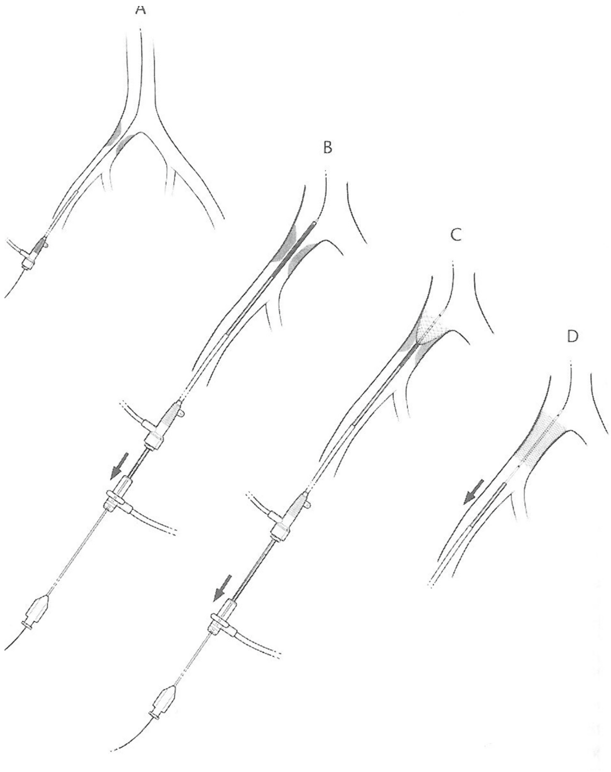


Fig. 2. Process of self-expandable stent placement [7]

process, we monitor the placement by X-ray machine. When pull back the catheter to do the removing of the catheter, the stent graft will expand staying still in the aorta aneurysm and remain fixed on aorta wall. When all of these done, we take back the guide line at last.

Differ from the self-expandable one, the balloon-expandable stent cannot do the placement by self-expanding. For that, we must use an expanding balloon helping to place the stent with the aid of pressure from aerating the balloon. Figure 3 shows how the work finishes.

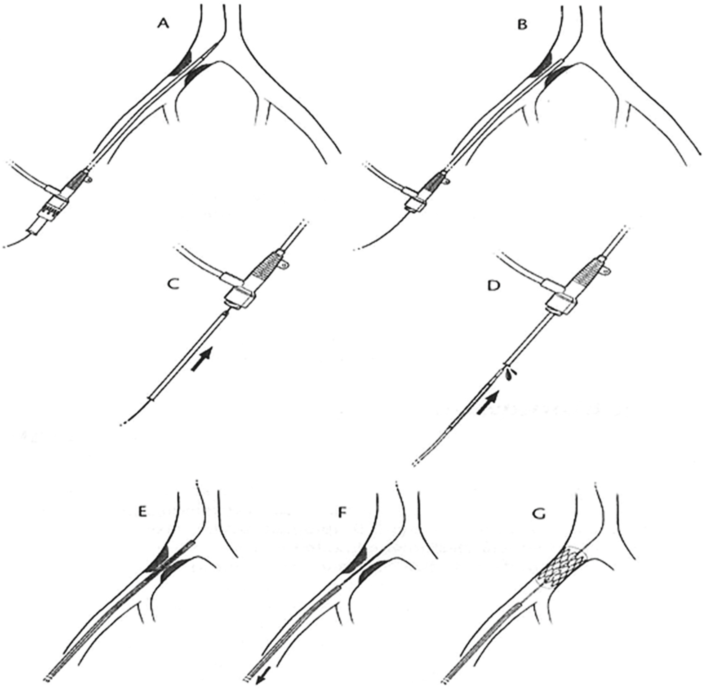


Fig. 3. Process of balloon-expandable stent placement [7]

2.3 Chronic Disease

According to the definition given by World Health Organization [8], Chronic diseases are diseases of long duration and generally slow progression. The more is that chronic occupies a large percentage in world major cause of death. A survey carried by World Health Organization showed that, during 2008, people who died of chronic occupied 63 % in the quantity of total world death. And in Taiwan 2012, eight seats was taken by chronic in the ten major cause of death including Cancer, heart disease, cerebrovascular disease, diabetes, chronic lower respiratory disease, hypertensive disease, chronic liver disease with cirrhosis and nephritis, nephritic syndrome and nephritis chronic diseases. So, chronic is a major cause of death not only in Taiwan, but also in the world.

2.4 Association Rule

Association rule is an algorithm which does an analysis focusing on the attributes of data and the association between items in data mining. For example, a database contains data of two attributes: A and B, a data has two items of A and B, and another data show that it also has items of A and B, then A and B may have certain association A → B. As a result, when A appears, B will appears follow A.

Association rule filters rules mainly based on Minimum Confidence and Minimum Support. Minimum Confidence represents the correct index of association rule. So the higher the Minimum Confidence shows, the more possible the rule is right. And the Minimum Support always shows the frequency that the association rule appears. In this case, when the support appears that means the rule often appears in database, but doesn't mean it is a right rule.

2.4.1 Apriori Algorithm

Aprioro algorithm was proposed by Agrawal [1] in 1994. In this algorithm, association rule make use of the earlier itemsets to calculate other related itemsets. If there's data package contains k items, we call this k-itemsets, and L_k means large k-itemsets as it shows in the Fig. 4 below. Form the figure, algorithm will find the L1 itemsets first and combine that as candidate L2 itemsets, then choose the support rate as the threshold value to screen out L2; after that, repeat the process to screen out L3. Till the end of this process that the algorithm cannot calculate any more, the program stops.

```
$L_1$ = {Large 1-itemsets};
**For** (k=2;$L_{k-1}\neq 0$;k++) **do begin**
    $C_k$=apriori-gen($L_{k-1}$);
    **For all** transactions t $\in$ **D do begin**
    $C_t$ = subset($C_k$,t);
    **For all** candidates c $\in$ $C_t$**do**
        c.count++;
**end**
$L_k$ = {c $\in$ $C_k$ | c.count $\geq$minsup}
**End**
Answer = $U_k L_k$;
```

Fig. 4. Apriori Algorithm [1]

3 Research Method

The circuit of this research is a modification of data mining circuit developed by FayyadU [3]. Figure 5 shows the circuit of this research step by step.

There are three steps in this research, the first is selecting the target data and setting fields value; the second is data preprocessing; and the third is data mining. After the algorithm finding out the association rule, it will be tested and verified. If the rule doesn't match with the requirement, the algorithm will back to any of the steps before, modify and find out the correct one.

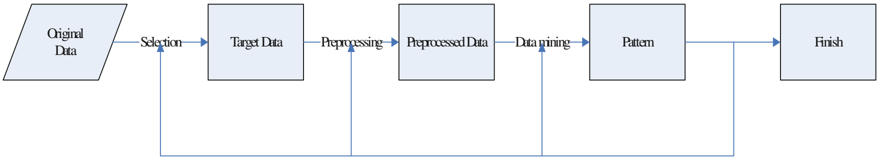


Fig. 5. Analysis flow diagram

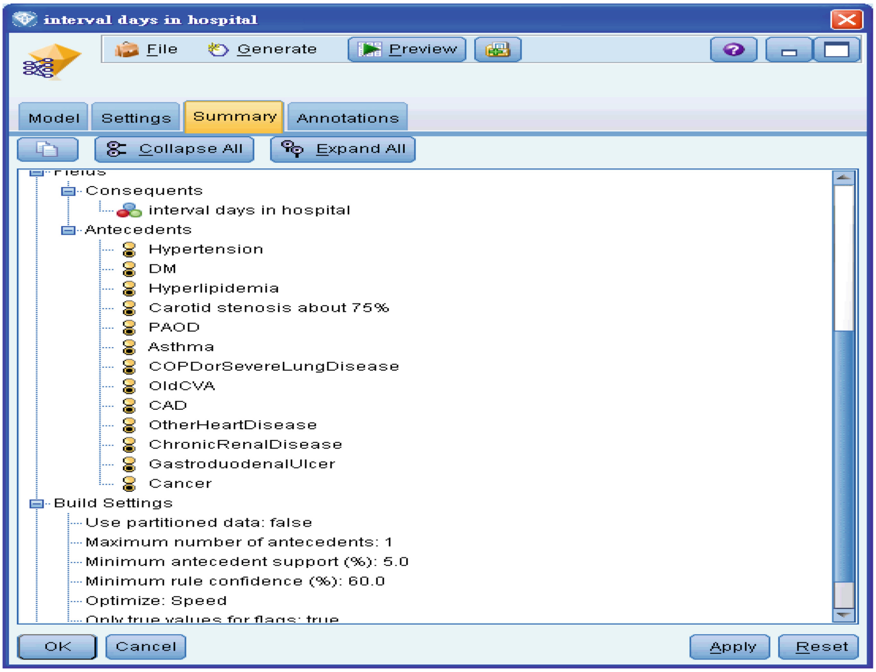
3.1 Analysis Procedure

The main research process is divided into three steps as follows:

A. **Select the target data and set the fields value**
The original data consists of a lot of different data, such as basic information of patient, medical history, data of operation, examination data and so on. However, examination data may shows the examinations of different time points, like scheduled examination after operation and periodic inspection back to the clinic. As a result, there will be more than one data containing operation data, data of examination before and after operation and data of periodic inspection back to the clinic besides the basic patient data. Each kind of data will make the patient data have over 100 different fields. In this research, the original data contains 184 patient data with over 100 fields in each. That means more than 10,000 fields of data needs to be analyzed and processed. However, not every field value is required by the research, we must learn the meaning of value in the original data after we get them so that we can take what we really want for the research. Otherwise, a lot of errors like incorrect value setting may mistake the algorithm or cause the inaccurate rule. That is why we should learn the data before we use them. After that, we shall continue set the data according to association rule and transform into proper value in order to let the algorithm run smoothly.

B. **Data preprocess**
After fetching the targeting data, we need to process them before insert into the algorithm. Because part of these data may have missing values, null values or incorrect values, and all of these may cause the algorithm run out inaccurate rule. For these reasons, we need to do preprocessing based on the first step.

C. **Data mining**
In order to reach the goal and find out the association between chronic and EVAR, we take the association rule algorithm of data mining developed by Apriori [1] to calculate the total 13 kinds of chronic in medical history data, and judge the recovery on basis of the days in hospital after operation. Table 1 is the set of interval days in hospital and 13 chronic (Figs. 6, 7 and 8).



```
Fields
  Consequents
    interval days in hospital
  Antecedents
    Hypertension
    DM
    Hyperlipidemia
    Carotid stenosis about 75%
    PAOD
    Asthma
    COPDorSevereLungDisease
    OldCVA
    CAD
    OtherHeartDisease
    ChronicRenalDisease
    GastroduodenalUlcer
    Cancer
Build Settings
  Use partitioned data: false
  Maximum number of antecedents: 1
  Minimum antecedent support (%): 5.0
  Minimum rule confidence (%): 60.0
  Optimize: Speed
  Only true values for flags: true
```

Fig. 6. Fields and parameter settings

Table 1. The set of interval days in hospital and 13 chronic

Interval days in hospital		Chronic
1–7 days	In one week	Hypertension
8–14 days	Between one week and two weeks	DM
15–30 days	Between two weeks and one month	Hyperlipidemia
Over 30 days	Over one month	Carotid stenosis about 75 %
		PAOD
		Asthma
		COPDorSevereLungDisease
		OldCVA
		CAD
		OtherHeartDisease
		ChronicRenalDisease
		GastroduodenalUlcer
		Cancer

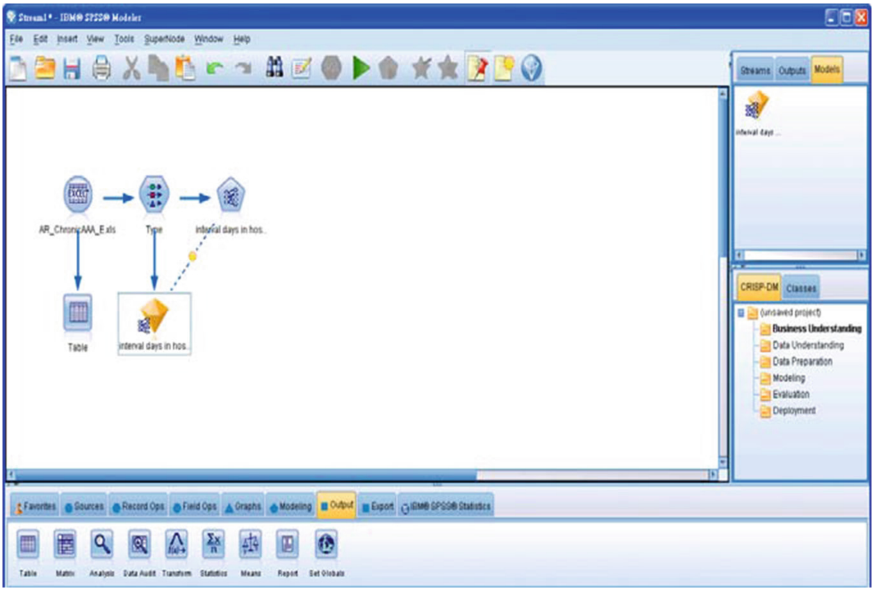

Stream1 * - IBM® SPSS® Modeler
File Edit Insert View Tools SuperNode Window Help
Streams Outputs Models
interval days ...
AR_ChronicAAA_E.xls
Type
interval days in hos..
Table
interval days in hos..
CRISP-DM Classes
(unsaved project)
Business Understanding
Data Understanding
Data Preparation
Modeling
Evaluation
Deployment
Favorites Sources Record Ops Field Ops Graphs Modeling Output Export IBM® SPSS® Statistics
Table Matrix Analysis Data Audit Transform Statistics Means Report Set Globals

Fig. 7. The mining architecture of this research

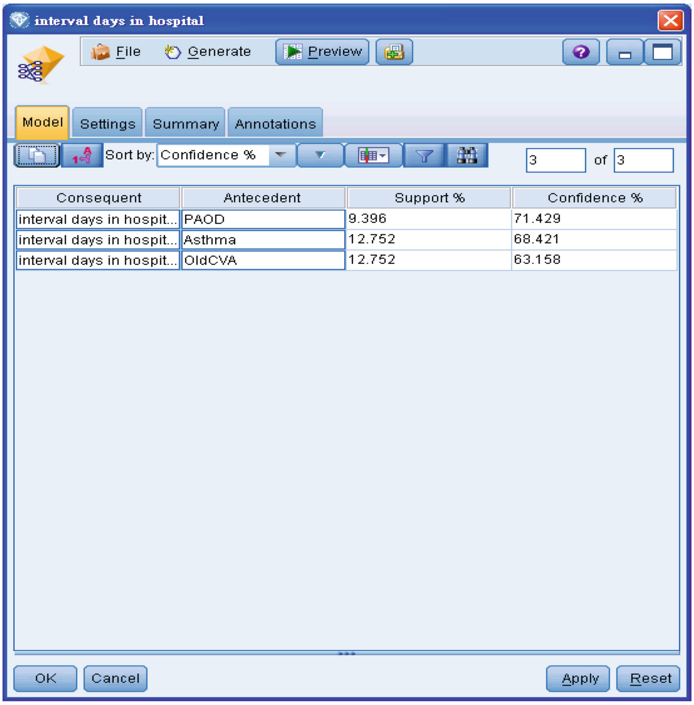
4 Research Result

From the original data, we fetched the chronic data in medical history as the target data, showed in Table 1. After data preprocessing, the missing values, null values or incorrect values were deleted. And then, we put the 149 patient data into association rule algorithm of data mining developed by Apriori [1] to calculate and analyze. Finally we set the Minimum Support at 60 % and got the following association rule:

PAOD	→	between two weeks and one month
Asthma	→	between one week and two weeks
OldCVA	→	between one week and two weeks

According to the rule, PAOD, Asthma and OldCVA may affect the recovery after operation. Patients who have these three chronic should spend at least one week, even one month to recover from the operation.

From the final result of this research, chronic may take the tendency to pose influence on patient after one week, however, the data of who leave the hospital in one week didn't show out the Support that high enough to affect the rule.



Sort by: Confidence %

3 of 3

Consequent	Antecedent	Support %	Confidence %
interval days in hospit...	PAOD	9.396	71.429
interval days in hospit...	Asthma	12.752	68.421
interval days in hospit...	OldCVA	12.752	63.158

Fig. 8. The final result of this research

5 Conclusion

During the research of original data, a lot of data still remain untreated, like medication records after operation or the level of endovascular aneurysm and some other data, all these data may affect the recovery. Therefore, the correctness and the accuracy of association rule will improve a lot if it is possible to collect the data of medical case of illness more completely and continue to analyze more patient data. Ultimately, Clinical diagnosis and treatment should be carried out by clinicians according to physical examination and other data. But this research can still provide a second reference for clinicians to treat the patients in advance, and improve the recovery after EVAR enabling them to leave hospital earlier and save more medical resource.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: 20th International Conference on Very Large Data Bases, (VLDB), pp. 498–499 (1994)
2. Barnes, M., et al.: A model to predict outcomes for endovascular aneurysm repair using preoperative variables. Eur. J. Vasc. Endovasc. Surg. **35**(5), 571–579 (2008)

3. Fayyad, U., et al.: The KDD process for extracting useful knowledge from volumes of data. Commun. ACM **39**, 27–34 (1996)
4. Ministry of Health and Welfare 2012 statistics of cause of death, 9 January 2014. http://www.mohw.gov.tw/EN/Ministry/Statistic.aspx?f_list_no=474&fod_list_no=4092
5. Moore, W.S., et al.: Abdominal aortic aneurysm: a 6-year comparison of endovascular versus transabdominal repair. Ann. Surg. **230**, 298 (1999)
6. Parodi, J.C., et al.: Transfemoral intraluminal graft implantation for abdominal aortic aneurysms. Ann. Vasc. Surg. **5**(6), 491–499 (1991)
7. Taiwan society for vascular surgery, 9 January 2014. http://www.tsvs.org/index.php
8. World Health Organization Chronic Disease, 9 January 2014. http://www.who.int/topics/chronic_diseases/en/

Human Action Recognition in Video Under Clutter and Moving Background

Der-Jyh Duh[1(✉)], Cheng-Chung Kan[2], Shu-Yuan Chen[2],
and Chia-Ming Lu[2]

[1] Department of Computer Science and Information Engineering,
Chien Hsin University of Science and Technology, Tao-Yuan 32097, Taiwan
djduh@uch.edu.tw
[2] Department of Computer Engineering and Science, Yuan Ze University,
Chung-Li 32003, Taiwan
s1006012@mail.yzu.edu.tw, cschen@saturn.yzu.edu.tw,
nekomio@gmail.com

Abstract. Behavior or action recognition in video sequences has becoming a more interesting and active research for computer vision. Applications of human action recognition, such as video retrieval, video surveillance and video event analysis are expanded extensively. Generally the video data used are based on fixed camera with stationary background, well-controlled environment and under simple actions. However, in real human action cases, the actions of human behavior are often complex and the background are cluttered with illumination changes, different human body size and moving camera. These make the real video much more complex. In this study, study to improve accuracies of human action recognition under clutter and moving background is proposed. The recognition scheme is based on the space-time interest point (STIP) and naïve Bayes based mutual information maximization (NBMIM). Methods including the selection of robust feature points based on camera motion estimation, analysis and correlations of the important STIP features during the training stage and weighting mechanism for action recognition to improve the recognition rate are used. Experimental results using the YouTube dataset indicate the effectiveness of the proposed scheme.

Keywords: Human action recognition · Spatio-temporal interest point · Camera motion estimation · Database clustering · Features correlation analysis

1 Introduction

The recognition of action in video sequences is an active and challenging area of research issue of computer vision, especially for the recognition of human action. Applications of human action recognition are widely developed, including video retrieval, video surveillance, video event analysis and human computer interaction. Conventionally human action recognition assumes that either the background in the video is stationary or the action of the human is simple. But for the database of real case, such as YouTube database and others, the camera moves along with the human so that the background is non-stationary. Besides, same action behaviors taken in real case

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 722–734, 2014.
DOI: 10.1007/978-3-319-13186-3_64

may not look like the same. So that descriptors between the different actions may not be similar. In this research, we propose a human action recognition system to improve the accuracy action recognition with complex contents and under low resolution.

1.1 Survey of Related Studies

Generally, approaches of recognizing human action can be divided into two categories, the top-down approach and bottom-up approach [1, 2]. Details of these two approached are given next.

(1) Top-Down approach:

Based on the characteristics of motion, motion or behavior descriptors are established and object recognition is made using the descriptors. These types of methods for human action recognition often use the image sequence instead of single image for recognition and generally take the entire human body as an object of interest. The appearance or movement trajectory of the human object is used as a descriptor for human motion identification. Based on the methods of object segmentation and descriptor for motion recognition, two categories can be roughly divided, (a) examplar-based model method and (b) state-based model method.

For the example-based method [3, 4], the recognition process is performed by using the video sequence of a complete human action. The video must be cut into a group of test unit first in order to calculate all the descriptors for each action and the similarity between the training and testing data is estimated. Since the speed of the action behavior for each person may not the same, descriptors obtained for the same action will not be exactly the same, constraints are needed when using these kinds of methods.

The state model-based methods [5–9] are based on the statistics or probability of action sequence. The action in the video sequence is represented by a series of sub-action model and a tree structure is constructed to represent the relations between these sub-action models. Dynamic Bayesian Network [5] and Hidden Markov Model [6] are well known model for calculating the probabilities of state transitions. The MEI (motion energy image) and MHI (motion history image) are proposed by Bobick and Davis *et al*. [7]. The idea of MHI is to construct a 2-D image as the descriptor. Each pixel in the MHI image represents the motion history of the same pixel in the video sequence, as shown in Fig. 1.

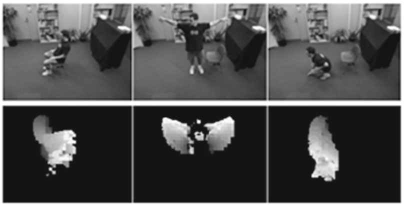


Fig. 1. Motion history image [9]

H.J. Seo *et al*. [3] and Wang *et al*. [4] development action recognition systems that use the behavior of human body contour as features. But the background is stationary and the contour of the human body must be accurately obtained, otherwise the recognition results are not accurate. Action recognition method based on the trajectory can overcome the problem when video is taken with change of view angles. KLT tracker or SIFT based tracker, such as Sheikh *et al*. [8] and Imran Junejo *et al*. [9], are commonly used for feature point tracking as shown in Fig. 2.

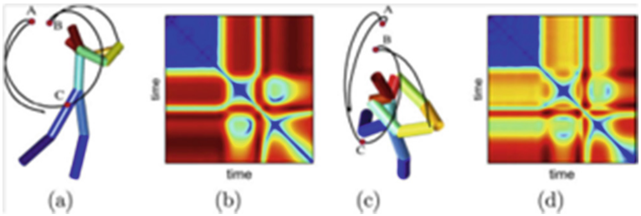


Fig. 2. Action recognition method based on the trajectory [9]

(2) Bottom-Up approach:

The bottom-up approach uses the feature derived from local spatial-temporal features and descriptors. These approaches take the human action as a set of descriptors from space-time point and use these spatial-temporal characteristics of the descriptor for action recognition. Local features from 2D images are extended to 3D video sequence and used to form so called as cuboids of video. The selection of cuboids generally is based on feature or control points from the image. Corner points with high brightness changes are most widely used. Harris detector is the most commonly used image corner point detection scheme. STIP is the most widely used feature point detection method as shown in Fig. 3. It is proposed by Laptev *et al*. [10] and the histogram of oriented gradient (HOG) and histogram of optical flow (HOF) are used as STIP descriptor. Other descriptors, such as 3D-HOG descriptor [11] and HOG-MHI descriptors [12] are also proposed by other researches.

Learning and classification algorithm are used for the STIP classification, such as SVM [10], Boosting [11]. Yuan *et al*. [13] proposed to use Naive-Bayes Mutual Information Maximization algorithm (NBMIM) for action recognition, Hongbo Zhang *et al*. [14] later proposed a scheme to improve its recognition rate. They used the ε-NN rule and the recognition results using KTH's database confirmed their idea.

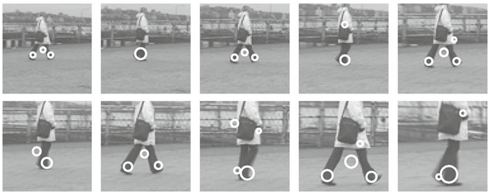


Fig. 3. Results of the local space-time feature point detection [12]

The proposed method is based on the method proposed by Hongbo Zhang *et al.* The benefits of our method are three-fold. First, in order to reduce the effect of moving camera which produces extra STIP on the background, a simple yet effective background motion decision scheme is proposed. Second, since the HOG feature is robust than the HOF feature, different weights are applied to the descriptors when calculating the descriptor distance in order to gain better recognition result. Third, by pre-clustering the library during the training phase, a weighted voting scheme is proposed to further improve the recognition accuracy. The function block of the proposed scheme is shown in Fig. 4.

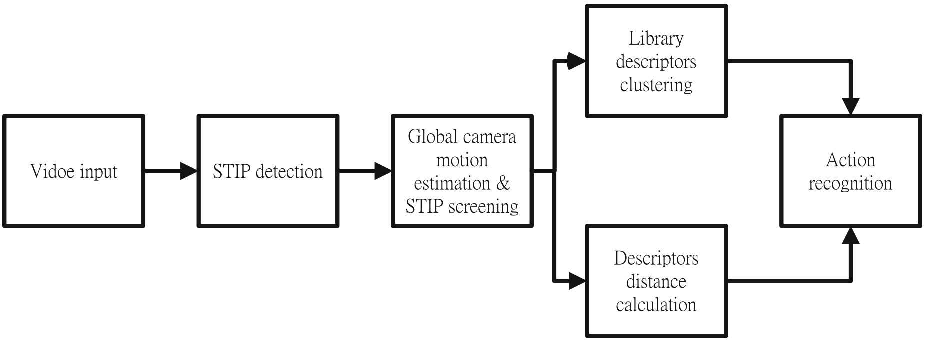

Vidoe input
STIP detection
Global camera motion estimation & STIP screening
Library descriptors clustering
Descriptors distance calculation
Action recognition

Fig. 4. The proposed system function blocks

The rest of the paper is organized as follows. Section 2 describes feature extraction and the descriptors used in this study. Section 3 discusses the proposed recognition scheme for humane action recognition. Experimental results are given in Sect. 4 and conclusion is drawn in Sect. 5.

2 Feature Extraction

2.1 FAST STIP Interest Point Detection

FAST corner detector is originally proposed by [15]. It is used for high speed corner detection and tracking of feature points in the video sequence. The basic idea of FAST corner point detection is that a test is performed for a feature at a pixel p by examining a circle of 16 pixels surrounding p. If the intensities of at least 12 contiguous pixels are all above or all below the intensity of p by some threshold, t, this feature p is defined as detected corner point. This type of interest point detection leads to using the pixel intensities from the circle as a descriptor. The benefits of using FAST interest point detection scheme are: (a) the computation is fast than that of standard STIP and (b) the repeatability of detected interest point is much better. So, instead of using the Harris detector, FAST STIP interest detection algorithm is used in this study.

2.2 Descriptor Construction

HOG was first proposed for human detection [10]. This method divides image windows into small spatial regions (called cells) first. A local histogram of gradient directions over the pixels of the cell is then accumulated for each cell. The most common method of gradient computation is simply applying the mask in one of or both the horizontal and vertical directions. In this study, two masks ([−1,0,1] and $[-1,0,1]^T$) are adopted to filter the color or intensity data of the image to obtain the orientation (or angle) of the current pixel. Then each pixel within the cell casts a weighted vote for an orientation-based histogram channel based on the values found in the gradient computation as shown in Fig. 5.

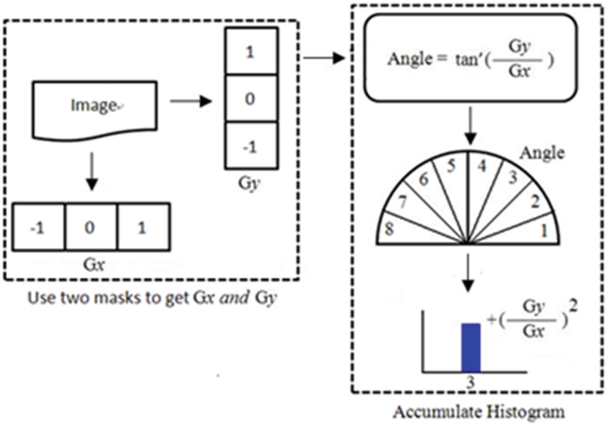


Fig. 5. Flow chart of HOG feature extraction

HOF calculates the dominant motion in each of the sub-regions. Both the amplitude and direction of motion are quantized through the use of 2D optical flow histograms, and therefore the dominant motion can be encoded simply by assigning a symbol to each of the histogram bins. This way, a compact representation of whole body motion is built. We call the sets of such symbol sequences HOF descriptors.

Kanokphan *et al.* [16] proposed a method to computing optical flow, the extracted silhouette centroid is positioned at the center of blank windows and the aligned window sequence is used to compute optical flow in the consecutive frames using Lucas-Kanade [17] algorithm. The direction of optical flow $\theta(x, y)$ is segmented into B regions which have the observation points on the circumference by ignoring optical flow magnitude. In our study, B is set as 16, θ is quantized into $\theta_k \in \{0^\circ, 45^\circ, 90, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ\}$. Direction histogram h_j in region j counts the number of observations which fall into each bin b_{jk}. The normalized histogram is created and motion vector for each frame is defined by concatenating direction histogram of optical flow in every region.

The descriptor for representing the interest point is made by concatenating the 72 dimension HOG vector with the 90 dimension HOF. So, the final descriptor for each interest point is a vector of 162 dimensions (Fig. 6).

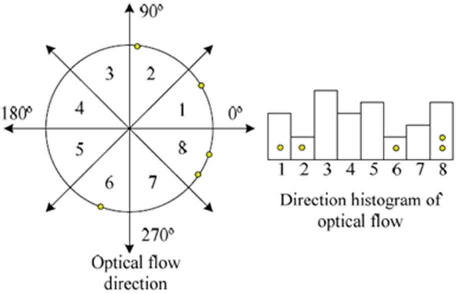


Fig. 6. Direction histogram of optical flow

3 Proposed Action Recognition Method

3.1 NBMIM Algorithm

The problem of action recognition is to identify the action that is performed in a given video sequence. Among various action recognition methods, the STIP method has achieved considerable success in recent years. In this thesis, a video is represented by a set of STIPs. The action recognition problem is then formulated as follows. Given a video sequence Q, a set of C action categories $\{1, 2, \cdots, C\}$, and a training set of labeled videos $T = \bigcup_{c=1,\cdots,C;n=1,\cdots,N} T_n^c$, identify the category of video. Here, N is the number of labeled videos for each category, so the training set T has $C \times N$ video sequences. In the STIP-based method, the video sequence Q is represented by a collection of STIPs that are detected from each frame of the video, and denote $Q = \{d_q\}$. Similarly, the training set T is denoted $T = \{d_t^c\} = \bigcup_{c=1,\cdots,C} T^c$ with $T^c = \left\{ d_t^c \middle| d_t^c \in \bigcup_{n=1,\cdots,N} T_n^c \right\}$.

The original conception about the action recognition using in this study is proposed by Hongbo Zhang, their method is based on NBMIM proposed by J. Yuan [13] for human action segmentation and classification. The recognition process is made by maximum voting score based on the mutual information which is estimated from the conditional similarity probability between the training model and testing model. The original NBMIM is based on the features of a collection of 3D-Harris interest points, which also represented by two spatial and temporal descriptors, HOG and HOF. The voting process uses STIP feature points and nearest neighbor (1-Nearest Neighbor (1-NN)) distance to estimate the probability density function.

Let a category C in database as positive samples, the positive class conditional probabilities of the STIP feature is $p\left(d_q \middle| \hat{c} = c\right)$. Let other categories of action as negative samples and the negative conditional probability of the STIP feature points is calculate as $p\left(d_q \middle| \hat{c} \neq c\right)$. The likelihood ratio of $\frac{p\left(d_q \middle| \hat{c} \neq c\right)}{p\left(d_q \middle| \hat{c} = c\right)}$ is used to determine if this STIP point is fall into positive or negative sample categories. For each STIP of the testing video sequence, if the STIP is classified as positive category, then a vote is added to the corresponding category C otherwise is it voted as against this action category C. After all the STIP feature points in the test video are process and all the votes are calculated, the category that get the most votes in the database is classified as the action of this test video. The calculation of the probability of the STIP and the value for voting determination is listed in Eq. (3).

$$\begin{aligned}\frac{p\left(d_q|\hat{c}\neq c\right)}{p\left(d_q|\hat{c}=c\right)} &= \frac{|T^{c-}|\sum\limits_{d_t^{\hat{c}}\in T^{c-}} K\left(d_q-d_t^{\hat{c}}\right)}{|T^c|\sum\limits_{d_t^c\in T^c} K\left(d_q-d_t^c\right)} \approx \frac{\exp\left[-\frac{1}{2\sigma^2}\left(\left\|d_q-d_{NN}^{c-}\left(d_q\right)\right\|^2\right)\right]}{\exp\left[-\frac{1}{2\sigma^2}\left(\left\|d_q-d_{NN}^{c}\left(d_q\right)\right\|^2\right)\right]} \\ &= \exp\left[-\frac{1}{2\sigma^2}\left(\left\|d_q-d_{NN}^{c-}\left(d_q\right)\right\|^2-\left\|d_q-d_{NN}^{c}\left(d_q\right)\right\|^2\right)\right] \\ \left\|d_q-d_{NN}^{c-}\left(d_q\right)\right\| &= \min_{d_t^{\hat{c}}\in NN_\varepsilon^{c-}\left(d_q\right)}\left\|d_q-d_t^{\hat{c}}\right\|, \left\|d_q-d_{NN}^{c+}\left(d_q\right)\right\| = \min_{d_t^{c}\in NN_\varepsilon^{c}\left(d_q\right)}\left\|d_q-d_t^{c}\right\|\end{aligned} \qquad (1)$$

Where$NN_\varepsilon^c\left(d_q\right)$ and $NN_\varepsilon^{c-}\left(d_q\right)$ is the number of STIP feature points that has descriptor distance less than ε with the sample data d_q in category c and category $\hat{c}\neq c$ respectively. $d_{NN}^{c-}\left(d_q\right)$ and $d_{NN}^{c}\left(d_q\right)$ are the nearest neighbors of d_q in the negative and positive datasets, respectively. The best probability distribution σ is selected by adjusting the value σ based on the purity in the neighborhood of the STIP d_q in NBMIM. $T^{c-}=\bigcup\limits_{\hat{c}\in\{1,2,\cdots,C\}\wedge\hat{c}\neq c} T^{\hat{c}}$ is the set of actions on behalf of all category C, d_q, $NN_\varepsilon^c\left(d_q\right)$ and $NN_\varepsilon^{c-}\left(d_q\right)$ are 128 dimensional vectors and the distance of two vectors used is Euclidean distance between two STIP feature point of HOG/HOF.

Hongbo Zhang *et al.* attempt to find a more effective way to obtain more STIP feature points in the training database during the training phase in order to enhance the accuracy of action recognition. They proposed to use two methods, one is take into account not only the most similar (nearest) single STIP feature point in the database but all the STIP feature points in a predefined distance range ε is used to calculate the conditional probability of similarity. In this way, with more training samples in database for voting, the recognition results should be better than simply using the 1-NN method. The proposed method is named as ε-NN and the estimation of the likelihood ratio is according to the following Eq. (4).

$$\begin{aligned}\frac{p\left(d_q|\hat{c}\neq c\right)}{p\left(d_q|\hat{c}=c\right)} &= \frac{|T^{c-}|\sum\limits_{d_t^{\hat{c}}\in T^{c-}} K\left(d_q-d_t^{\hat{c}}\right)}{|T^c|\sum\limits_{d_t^c\in T^c} K\left(d_q-d_t^c\right)} \approx \frac{\left|NN_\varepsilon^{c-}\left(d_q\right)\right|\sum\limits_{d_t^{\hat{c}}\in NN_\varepsilon^{c-}\left(d_q\right)} K\left(d_q-d_t^{\hat{c}}\right)}{\left|NN_\varepsilon^{c}\left(d_q\right)\right|\sum\limits_{d_t^{c}\in NN_\varepsilon^{c}\left(d_q\right)} K\left(d_q-d_t^{c}\right)} \\ &\approx \frac{\left|NN_\varepsilon^{c-}\left(d_q\right)\right|\sum\limits_{d_t^{\hat{c}}\in NN_\varepsilon^{c-}\left(d_q\right)} \exp\left[-\frac{1}{2\sigma^2}\left(\left\|d_q-d_t^{\hat{c}}\right\|^2\right)\right]}{\left|NN_\varepsilon^{c}\left(d_q\right)\right|\sum\limits_{d_t^{c}\in NN_\varepsilon^{c}\left(d_q\right)} \exp\left[-\frac{1}{2\sigma^2}\left(\left\|d_q-d_t^{c}\right\|^2\right)\right]}\end{aligned} \qquad (2)$$

3.2 Proposed Scheme

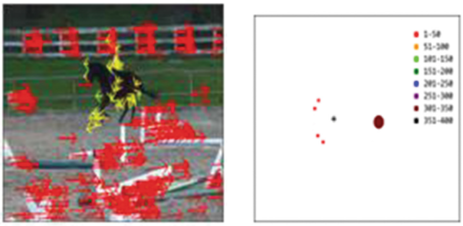
Hongbo Zhang's action recognition scheme cannot accurately apply to the YouTube video data with dynamic and complex background. There are several reasons; first, due to the dynamically moving background in the video, the detected STIP feature points are very unstable and unreliable. Both the background and the human foreground contain STIP feature points. Furthermore, the calculated HOF will not be precise due to the motion of camera and the HOG/HOF descriptor will be unreliable which makes the action recognition with poor results.

In this study, three mechanisms are proposed to improve the performance of Hongbo's scheme when applied to the YouTube database. The process includes: (1) camera motion vector estimation and interest point screening, (2) action database correlation and clustering analysis and (3) weight used for descriptor in action recognition. The details are given in the followings:

(1) Camera motion vector estimation and interest point screening:

With the moving background, the extracted interest feature points may cover over the human and the background. So, a screening process is needed in advance to prune out all the possible interest feature points of the background. The global camera motion estimation is estimated first and those corner points with the similar motion vector will be assume to belong to the background and will all be deleted. A quantized 2-D polar motion vector histogram of each frame is calculated. The motion vector with the highest histogram count is assumed to be the background motion. Those corner points with different motion vector are referred as the interest point of the human.

The direction of the optical flow calculated for each interest point is quantized into 18 bins and the strength of the optical is quantized into 3 bins. The accumulated histogram of all the optical vectors of the interest point is calculated as shown in the following figure. To each bin in the histogram, the count of the four neighbors is also added to avoid the error cause by quantization. The red vectors shown in Fig. 7(a) are the background corner points with its motion vector. The vectors in yellow color are the STIP corner points after screening. The results are quite satisfactory. Figure 7(b) shows the results of numbers of corner points after 4-neighbour voting considering where the position of the big brown dot is the estimated camera motion vector.



(a) (b)

Fig. 7. (a) The 2D interest point corner with its optical flows motion vector (b) Accumulated histogram of STIP motion vector.

(2) Action database correlation and clustering analysis:

Since the interest descriptors are noisy, the degree of clustering or correlation of the interest descriptors should be estimated. If some of the interest descriptors for the same action library are very similar, that is they are highly correlated, the discrimination ability of these interest features vector should be better than the others and they are defined as high-discrimination cluster. So, during the recognition phase, the distance weight of these interest descriptors should be set greater than the others. The number of the cluster for each action is calculated adaptively by the distribution of the distance between each descriptor. Example using 3 clusters is shown in Fig. 8.

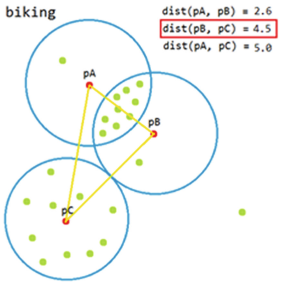


Fig. 8. Scheme of clustering and distance calculation with 3 clusters.

(3) Weight of descriptor used in action recognition:

Since the HOG feature is much unreliable, the distance between two descriptors using NBMIM need to be modified. Instead of using the same weights as in NBMIM, different weights are applied to HOG and HOG. Equations for calculating the distance are shown in the followings and the definitions of each parameter used are also listed (Table 1).

$$\mathrm{D_G} = \sum\nolimits_{\mathrm{i}=0}^{71} \sqrt{(F_A[\mathrm{i}] - F_B[\mathrm{i}])^2} \tag{3}$$

$$\mathrm{D_F} = \sum\nolimits_{\mathrm{i}=72}^{161} \sqrt{(F_A[\mathrm{i}] - F_B[\mathrm{i}])^2} \tag{4}$$

$$\mathrm{T} \;= \mathrm{average}\left(\exp^{\left(negSig*\sqrt{W_G D_G^- + W_F D_F^-}\right)}\right)/\mathrm{verage}\left(\exp^{\left(negSig*\sqrt{W_G D_G^+ + W_F D_F^+}\right)}\right) \tag{5}$$

$$\frac{p(d_q|\hat{c} \neq C)}{p(d_q|\hat{c} = C)} = W_L * \log\left(\frac{C}{1 + (T * (C - 1))}\right) \tag{6}$$

The *negSig* is for calculating the probability distribution of mutual information. C is the number of action class. W_G and W_F is the weighting factor used when calculating difference between HOG and HOF descriptor. W_L is the weighting factor for the three different types of cluster and the value is set experimentally.

Table 1. Definition of parameters

Parameter	Definition
negSig	Standardized constant (−3.125)
C	The number of categories (11)
W_G	Weight of HOG
W_F	Weight of HOF
W_L	Weight of different cluster
D_G	Euclidean distance of HOG
D_F	Euclidean distance of HOF

4 Experimental Results

The proposed method was implement on PC with a 3.4 GHz core Intel i7-3770 CPU and 16 Gigabytes DDR3SDRAM. The YouTube video database used in this study contains 11 different action categories and with a total of 1599 videos. Through this study, 12 tests are conducted using one third of the video randomly selected in each category as testing videos and the left two thirds videos as training videos.

1. Camera motion vector estimation and interest point screening:

Some of the results of interest point detection and screening are shown in the next figure. We can easily find that the final interest points (colored in yellow) used for further action recognition are quite satisfactory. Those interest points belong to the background (colored in red) are correctly detected.

Only a very small part of the interest point of human are classified as background point. This makes only slightly effect on the action classification which is based on the whole interest points of the human. An average of 86.3 % of correctly screening rate is achieved with only about 8.5 % of error rate (Fig. 9).

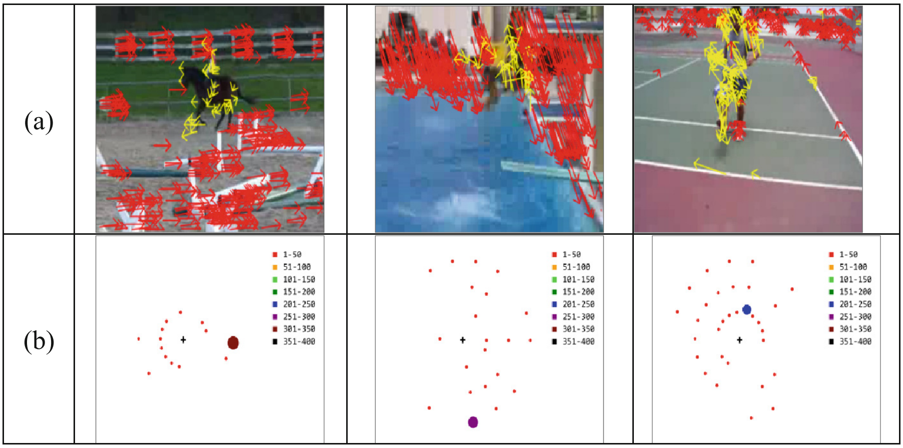


Fig. 9. (a) Detected interest points (red color: background part, yellow color: human part) (b) The motion vector histogram and the detected background motion vector (the largest spot) (Color figure online).

Table 2. Recognition rate of using different W_G

W_G	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	3.0
Average	36.4	36.4	45.5	45.5	36.4	36.4	36.4	36.4	36.4	36.4

Table 3. Recognition rate of using different W_L

W_L	1.25	1.50	1.75
Average	48.5	51.6	48.5

Table 4. Confusion matrix

	riding	**swing**	**biking**	**shooting**	**diving**	**golf**	**juggle**	**tennis**	**jumping**	**spiking**	**walking**
riding	**91.7**	8.3									
swing		**75.0**	8.3		8.3				8.3		
biking	16.7		**75.0**	8.3							
shooting		16.7		**83.3**							
diving	50.0	8.3		8.3	**33.3**						
golf	25.0					**58.3**		8.3	8.3		
juggle		16.7	8.3		8.3	8.3	**50.0**	8.3			
tennis	8.33	8.3			8.3		8.3	**50.0**	16.7		
jumping	41.7	8.3	8.3		8.3			16.7	**16.7**		
spiking	33.3	25.0					8.3			**33.3**	
walking	16.7	16.7	8.3					16.7			**41.7**

2. Weight analysis of descriptor used inaction recognition:

First of all, three experiments are made to test the effect of selecting different HOG. We select different value of WG and action recognition is performed without using the library action cluster scheme. The average recognition rate is shown in Table 2. From the testing result, the value of WG is set as 2.4 in this study.

A second test is conducted to check the effect of selecting different weight for library cluster. We set the weight of high-discrimination cluster to 2, the weight of non-cluster to 1 and the weight of normal-discrimination cluster with three different values is used. The average recognition result is shown in the next Table. From the result, we set the weight of normal-discrimination cluster to 1.5 is this study (Table 3).

The confusion matrix is shown in the next Table. It is noted that due to the similarity of different action, some of the result may not in very good condition. An average of 55.3 % of recognition rate is achieved, with an improve of 5 % comparing to the results of [17] where the action result for the YouTube database is about 50 % (Table 4).

5 Conclusions

A robust human action recognition scheme under clutter and moving background conditions is proposed in this study. The recognition scheme is based on the STIP and NBMIM. Methods including the selection of robust feature points based on camera motion estimation, analysis and correlations of the important STIP features during the training stage and weighting mechanism for action recognition for improving the recognition rate are used. Experimental results using the YouTube dataset indicates the effectiveness of the proposed scheme.

Acknowledgments. This work was partially supported by National Science Council of Taiwan, under Grants NSC-101-2221-E-231-029.

References

1. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: a review. ACM Comput. Surv. **43**(3), 16 (2011)
2. Roppe, R.: A survey on vision-based human action recognition. Image Vis. Comput. **28**(3), 976–990 (2010)
3. Seo, H.J., Milanfar, P.: Action recognition from one example. IEEE Trans. Pattern Anal. Mach. Intell. **33**(5), 867–882 (2011)
4. Wang, Y., Mori, G.: Human action recognition by semi-latent topic models. IEEE Trans. Pattern Anal. Mach. Intell. **31**(10), 1762–1774 (2009)
5. Park, S., Aggarwal, J.K.: A hierarchical Bayesian network for event recognition of human actions and interactions. Multimedia Syst. **10**(2), 164–179 (2004)
6. Natarajan, P., Nevatia, R.: Coupled hidden semi Markov models for activity recognition. In: Proceedings of the IEEE Workshop Motion and Video Computing (2007)
7. Bobick, A., Davis, J.: The recognition of human movement using temporal templates. IEEE Trans. Pattern Anal. Mach. Intell. **23**(3), 257–267 (2001)
8. Sheikh, Y., Sheikh, M., Shah, M.: Exploring the space of a human action. IEEE Trans. Pattern Anal. Mach. Intell. **33**(1), 172–185 (2011)
9. Junejo, I., Dexter, E., Laptev, I., Perez, P.: View-independent action recognition from temporal self-similarities. In: Proceedings of the IEEE International Conference on Computer Vision (2005)
10. Laptev, I.: On space-time interest points. Int. J. Comput. Vis. **64**(2–3), 107–123 (2005)
11. Kläser, A., Marszałek, M., Schmid, C.: A spatio-temporal descriptor based on 3D-gradients. In: Proceedings of the British Machine Vision Conference (2008)
12. Tian, Y.L., Cao, L.L., Liu, Z.C., Zhang, Z.Y.: Hierarchical filtered motion for action recognition in crowded videos. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **42**(3), 313–323 (2012)
13. Yuan, J., Liu, S., Wu, Y.: Discriminative video pattern search for efficient action detection. IEEE Trans. Pattern Anal. Mach. Intell. **33**(9), 1728–1743 (2011)
14. Zhang, H.B., Li, S.Z., Su, S.Z., Chen, S.Y.: Selecting effective and discriminative spatio-temporal interest points for recognizing human action. IEICE Trans. Inf. Syst. **E96-D**(8), 1783–1792 (2013)

15. Rosten, E., Porter, R., Drummond, T.: Faster and better: a machine learning approach to corner detection. IEEE Trans. Pattern Anal. Mach. Intell. **32**, 105–119 (2010)
16. Lertniphonphan, K., Aramvith, S., Chalidabhongse, T.: Human action recognition using direction histograms of optical flow. In: ISCIT (2011)
17. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: DARPA Imaging Understanding Workshop, pp. 121–130 (1981)

Big Data Generation: Application of Mobile Healthcare

Huan-Chao Keh[1], Lin Hui[2(✉)], Kuang-Yi Chou[3],
Yuan-Cheng Cheng[1], Po-Yuan Yu[1], and Nan-Ching Huang[1]

[1] Department of Computer Science and Information Engineering,
Tamkang University, New Taipei City, Republic of China
{Keh, 144920, 132310}@mail.tku.edu.tw,
fishyu750502@hotmail.com
[2] Department of Innovative Information and Technology, Tamkang University,
Yilan County, Republic of China
121678@mail.tku.edu.tw
[3] Center of General Education Taipei City,
National Taipei University of Nursing and Health Sciences,
Taipei, Taiwan (R.O.C.)
kuangyi@ntunhs.edu.tw

Abstract. Taiwan has been entering an aging society, with the issue of taking care of the elderly being more and more severe. Although many home care systems have come out being sold in the market, there's still a long way to develop a Tele-health care system in order to let the elderly use it efficaciously away from their home. The approaching of massive data era drives the cloud computing being popular, and our government also take action to integrate the database of National Health Insurance Research, establishing The Big Data for Health Care. All they want is to bring more benefits to health care. This research take purpose to propose an early warning process with return of initiative judging whether the device return correctly; we will construct a micro mobile care system as well, judging users' health condition efficaciously by association rule, completing the timely informing, helping the elderly to achieve a better life, reducing the waste of medical resource.

Keywords: Tele-health care · Mobile care · Data mining

1 Introduction

After the national health care carried out in Taiwan, the department of national health insurance constructed a system which record the countrywide nationals' health condition and medical treatment. Since the database of National Health Insurance Research was established by NIH in Taiwan 1998, it has been in use for over 16 years (1998–2013) consists a great size of 23 million records and it is also full of rich worthy mining potential data not only for curing diseases, but also for the effect of treatment. In 2013, the executive office of science and technology reporting has devote almost one billion to construct Big Data for Health Care for health care, disease preventing and researching, as well as new medicine developing. In the future, the database of national

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 735–743, 2014.
DOI: 10.1007/978-3-319-13186-3_65

insurance research and Taiwan gene research will be integrated and become a larger value-added database, for each year, it can reduce NT 100 billion consuming of health care. [1].

Since 1993, Taiwan began to entering an aging society, people who over 65 years old would have a significant increase. Till 2060, the number will reach to 7,460,000 (Fig. 1) almost the three times of 2011. At the same time, the labor force of people who between 15–64 years old just remain 9,600,000 about 55.8 % of 2011, and that will put a huge pressure on taking care of the needed [2]. As we all know, the elderly is easier to suffer from chronic like Hypertension, high cholesterol and some others due to the age and life style, which will make them a heavier burden to be cared. Therefore, the Tele-health care comes to be another solution to the issue of aging society in Taiwan.

This research aims to develop a micro mobile care timely initiative warning system to satisfy the care in distance, inform the liaisons and medical units which have been set ahead of time when the emergency happened to the user; the mobile device will transfer the data like heart beating, breath, body temperature, acceleration, gyroscope and GPS value, from sensor to remote serve and take down through Wireless Wearable Body Area Network (WWBAN). The situation of user being in danger can be forecasted by the association rule which is inserted ahead of time. Assuming that each user uploads 10 MB data each day, when the users reach 1000 each day, 1 GB data will continue to add into server. If we take a long time to trace the user to produce some meaningful clinic rule, huge amount of physiological data would need a platform superior to store. The present information collecting of once per second will increase the device power consuming. However, if we shift to the server side and collect information by server judgments, that can increase the working time of the device and improve the rule to be more accurate.

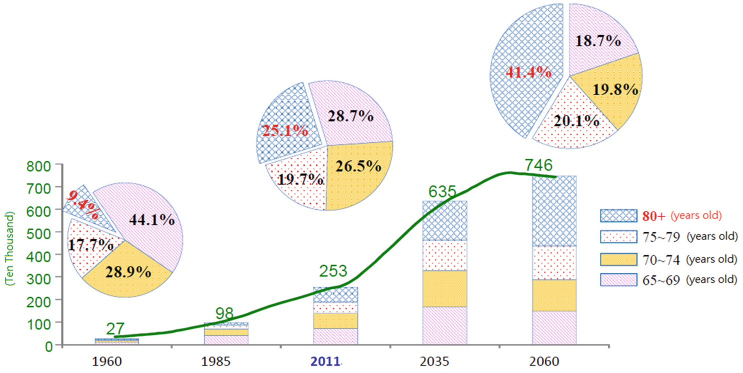


Fig. 1. Taiwan population structure (Source: Council for Economic Planning and Development, R.O.C 2012)

2 Related Work

2.1 Big Data

In the modern time of web2.0, social network has become another activity people are familiar with, and 2.5 EB (2^1018 Bytes) of new data will be produced every day by uploading messages, photos and film file. Such a huge amount of data has attracted the crowds' attention, and it is forecasted that 50 times of data would born till 2020 compared nowadays. There are 4.6 billion mobile phones in use throughout the world because of the big bang of mobile devices. It has become more and more important to deal with the response for that 0.1–0.2 billion people are active online.

A lot of useful application can be created from these data, in word processing, questions can be answered by collecting messages; in business analysis, clients can be deeply learned, as well as the decision analysis in business intelligence; in online analysis, joint data mining and social recommendation can be carried out; even the analysis of human behavior, mood, study and health can be carried out from the huge data.

Huge data is less structural, dispersion and complex. Usually they have the feature of 3 V: Volume, Variety and Velocity. The data of patients collected by medical department includes medical history in words, all kinds of value of examination like X-ray, MAI and SONO, and some other clinic information like Continuous observation of Vital Signs. Therefore, evidence-based medicine can be another field for cloud computing application.

2.2 Association Rule

Data mining is one step in the process of knowledge development, and association rule is one method to do the data mining [6], with the most famous application of Wal-Mart shopping records association rule, such as people who bought the diapers also bought beers and enhance the turnover. This shopping analysis can reach the goal of forecasting analysis. In 2012, New York Times reported a news <How Companies Learn Your Secrets> [7], which described a story that a father was shocked by Target market who knew the information that his daughter was pregnant earlier than him.

This is so called the forecast of association rule. USA Target market use the GUEST ID to connect each behavior of customer, including questionnaire, credit card consumption or the record of using coupons, even the behavior of calling the customer service hotline and surfing the mal website. Through all these data and combine the information of age and gender, a forecasting model of the pregnant comes out. That means once the customer buy the good match well with the model, the information of special offers would be sent out automatically. Especially for this, we know that data mining is a good way to forecast.

The Tele-health care system differs from the traditional one in the fields that users use it without environmental limitation. Therefore, the system can just judge the condition through the instant data transferring. Besides the basic data inserted into the system at the very beginning, the system also need to receive the data from client sensor and respond immediately.

3 Method

3.1 System Design

User wear sensor strap on own trunk and used the intelligent phone to connect sensor as well as monitor server. From Fig. 2 we can know that user has to login at first to make made sure who is right user. In “Setting”, the main function was to set the emergency contact and manual adjust vital sign detect threshold value which was download from server for real time detect user’s physical change. On the phone screen, user not only can see self-physical condition real time which sensing data transmits from sensor by Bluetooth, but also show that medical unit’s health care notice.

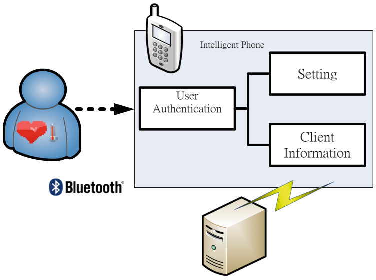


Fig. 2. Intelligent phone function

In server-side (Fig. 3), this study decomposes it to three main modules and two databases. Data collect module has two units: one was network communication unit, another was data process unit. Network communication unit was connect to user’s intelligent phone through internet (Wi-Fi or 3G signal) to transmit user’s association rule to client side; also received all data from sensor such as heart rate, respiration rate, body temperature, posture and acceleration values.

Data process unit is a filter to filter out instant abnormal signal, for example a series of heart rate average 90, suddenly one signal show as 250 was an electronic signal error, and the filter will ignore the signal. Finally, it transfers these data to physical signal database through API and monitor module as same time.

Monitor module were most important module. Because this module was responsible for compare users’ real time data with association rules which from rules database, for example one rule was:

$$[\text{Posture degree} < -60 \text{ and } \text{Acceleration value} > 5] - > [\text{fall down}].$$

That posture degree means Backwards degree count from stand straight; the acceleration value > 5 means the speed quickly than usual walking.

And user report a data was [posture degree: −85, acceleration value: 5.5]. Physical condition judgments unit compare this data with rule will be and emergency situation that emergency call unit to start actively alert flow which introduce in next section. Health records module provide main control and monitor screen, patients’ family or

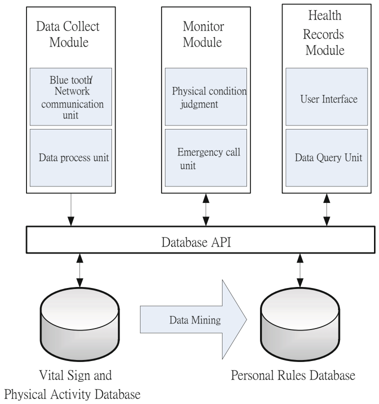


Fig. 3. Server side module

medical unit can monitor patient's physical change and query history data. Furthermore, Doctor or Nurse can give medical order or suggestion to patient.

3.2 Actively Alert Flow

Although that we set up the threshold value to reduce probability of error of judgment; but sometimes still have error estimation that cause intelligent phone send a help message. So we have design the process of actively alert flow that verify that the user whether have problem in actually. Such as Fig. 4, we dial the automatically voice calls to user to confirm conscious of user, once the phone call be answer, the system will record the user and background sound. When the user presses the number 1 indicates who is safe now; if the user presses the number 2 indicates that the user is conscious, but maybe feels uncomfortable and needs help. The system will according to the different set up communication channel to transmit message to the specified object to the rescue. Press the number 3 system will open the user and the two designated contact person (such as family members and attending physicians) be a three-way calling. Once that calls to user and no answer in three times, it will automatically start rescue process.

3.3 Data Mining

In the research, we used IBM DB2 Miner to generate association rule. Every data we collected from sensors all as continue data, if we want to use association rule technique to generate association rule of physical judgement condition, we have to transfer these data to nominal data format. Such as that heart rate > 80 means" quick", between 80 and 60 means" normal", lower than 60 was "slow". We expect that through simulation different posture combine physical data change which can get some rules for practical application.

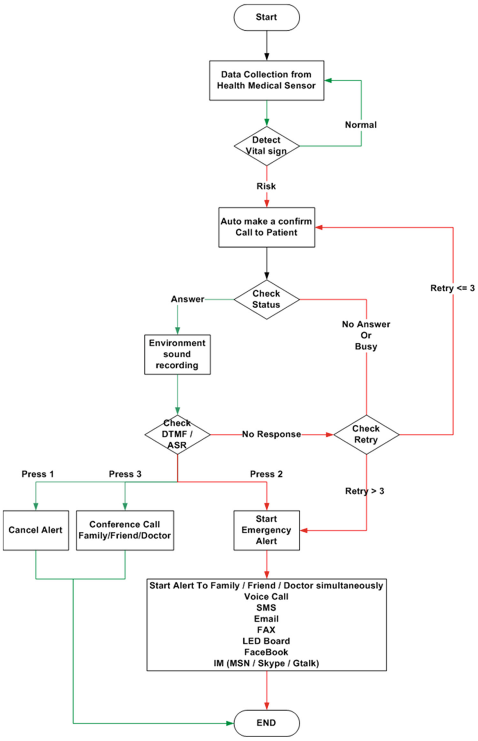
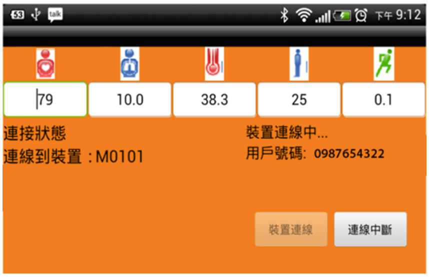


Fig. 4. System alert flow

4 Result and Discussion

According to our design, the development platform was Android system. This figure show out user interface and all real time detected object include heart rate, respiration rate, body temperature, user posture and acceleration value (Fig. 5). User can connect or disconnect to sensor through the buttons on the lower right corner of screen.

Although that we were still mining association rule, monitor can manual set up threshold value for physical detection on setting page in phone.



79 10.0 38.3 25 0.1

連接狀態
連線到裝置 : M0101

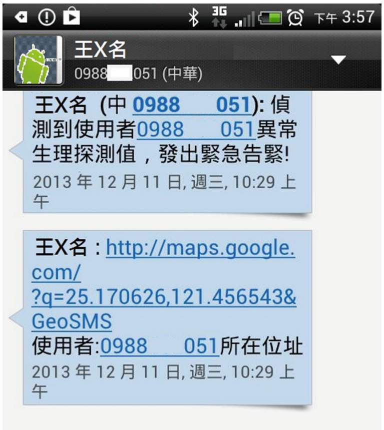
裝置連線中...
用戶號碼: 0987654322

Fig. 5. User interface of android system

We applied the system to a 57-year-old man with history of hypertension and chronic high blood lipids. He needs to take blood pressure-control drugs regularly. The patient uses smartphone to access the Internet by 3G. His mobile phone number was set as the system user ID. Meanwhile, the researcher was set as his designated contact person.

The user experiment was conducted for a month from December, 2013. During the experiment, wherever the patient went, he would have to carry the mobile phone with him. Though there were 10 error returns happened during the process, the patient eventually fixed the errors by using the voice instruction system.

On December 11th, the researcher received four types of emergent messages from the system. One contained alarm message and the user's location via the user's mobile

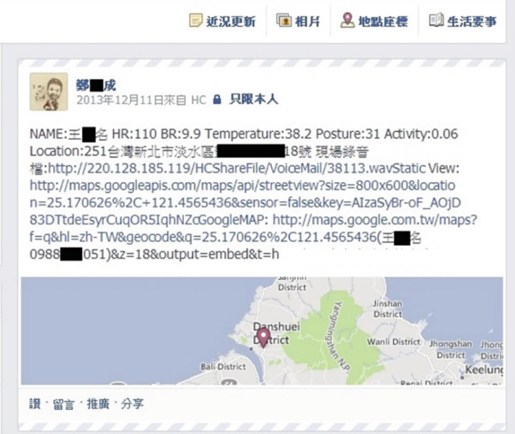


王X名
0988[redacted]051 (中華)

王X名 (中 0988[redacted]051): 偵測到使用者0988[redacted]051異常生理探測值，發出緊急告緊!
2013 年 12 月 11 日, 週三, 10:29 上午

王X名：http://maps.google.com/?q=25.170626,121.4565438&GeoSMS
使用者:0988[redacted]051所在位址
2013 年 12 月 11 日, 週三, 10:29 上午

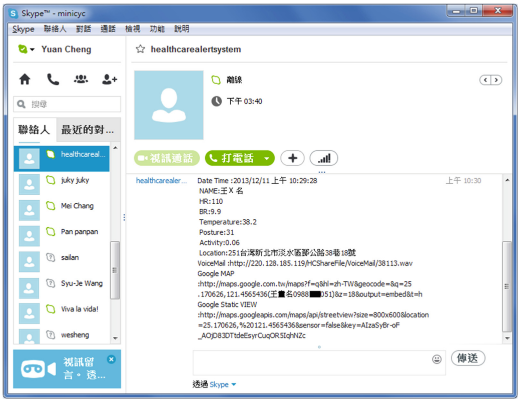
Fig. 6. APP sent SMS to contact



鄭[redacted]成
2013年12月11日來自 HC 只限本人

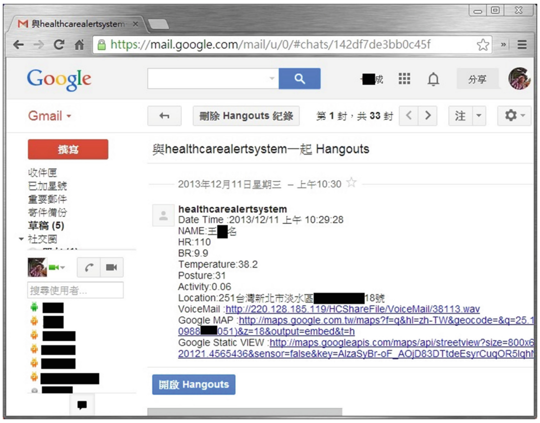
NAME:王[redacted]名 HR:110 BR:9.9 Temperature:38.2 Posture:31 Activity:0.06 Location:251台灣新北市淡水區[redacted]18號 現場錄音檔:http://220.128.185.119/HCShareFile/VoiceMail/38113.wavStatic View: http://maps.googleapis.com/maps/api/streetview?size=800x600&location=25.170626%2C+121.4565436&sensor=false&key=AIzaSyBr-oF_AOjD83DTtdeEsyrCuqOR5IqhNZcGoogleMAP: http://maps.google.com.tw/maps?f=q&hl=zh-TW&geocode&q=25.170626%2C121.4565436(王[redacted]名 0988[redacted]051)&z=18&output=embed&t=h

Fig. 7. Server node sent user's location and voice recording to FaceBook



healthcarealer... Date Time :2013/12/11 上午 10:29:28 上午 10:30
NAME:王X 名
HR:110
BR:9.9
Temperature:38.2
Posture:31
Activity:0.06
Location:251台灣新北市淡水區鄧公路38巷18號
VoiceMail :http://220.128.185.119/HCShareFile/VoiceMail/38113.wav
Google MAP
:http://maps.google.com.tw/maps?f=q&hl=zh-TW&geocode=&q=25.170626,121.4565436(王[redacted]名0988[redacted]051)&z=18&output=embed&t=h
Google Static VIEW
:http://maps.googleapis.com/maps/api/streetview?size=800x600&location=25.170626,%20121.4565436&sensor=false&key=AIzaSyBr-oF_AOjD83DTtdeEsyrCuqOR5IqhNZc

Fig. 8. SKYPE user's information to contact



與healthcarealertsystem一起 Hangouts

2013年12月11日星期三 – 上午10:30

healthcarealertsystem
Date Time :2013/12/11 上午 10:29:28
NAME:王[redacted]名
HR:110
BR:9.9
Temperature:38.2
Posture:31
Activity:0.06
Location:251台灣新北市淡水區[redacted]18號
VoiceMail :http://220.128.185.119/HCShareFile/VoiceMail/38113.wav
Google MAP :http://maps.google.com.tw/maps?f=q&hl=zh-TW&geocode=&q=25.1
0988[redacted]051)&z=18&output=embed&t=h
Google Static VIEW :http://maps.googleapis.com/maps/api/streetview?size=800x6
20121.4565436&sensor=false&key=AIzaSyBr-oF_AOjD83DTtdeEsyrCuqOR5Iqh

Fig. 9. Pass to contact by Hangouts

phone side which is called as GeoSMS system (Fig. 6). One contained live recordings, physiological data, and location data on Google map (Fig. 7). The server side sent all information to the designated contact person's Facebook. Figure 8 shows the server sent alarm message to Skype. Figure 9 shows the server sent the message to the Hangouts. (Contact person can enter the user ID to get the latest abnormal returns by Hangouts.)

The moment the contact person receives the emergent messages, he calls the patient/user immediately. The patient said that he forgot to take high blood pressure-control medicine, so he got stuck in a discomfort situation. Through the experiment results, we could prove the system indeed achieves its instant notification effect.

5 Conclusion

Most of long-distance health care systems focus on indoor care, this study integrated micro sensor, intelligent phone and monitor system to make sure patient's outdoor safety, and we finished initial stage about sensor with intelligent phone communication. In additional, we collected patients' physical and activity data to find their own association rule through data mining which fit every patient actual physical condition. For instance, the standard adult heart rate was between 60 ~ 80 where was calm down. But, in special case who's heart was between 50 ~ 60 in clinical. So we from these data can generate the rule fit for the patient. Finally, this study can customize detection rules for patients that have high quality life.

References

1. WiKi. http://zh.wikipedia.org/wiki/%E8%B6%85%E7%B4%9A%E5%81%A5%E5%BA%B7%E9%9B%B2
2. http://www.cepd.gov.tw/m1.aspx?sNo=0000455

3. Saracci, R., Samet, J.: Commentary: Call me on my mobile phone…or better not?—a look at the INTERPHONE study results. Int. J. Epidemiol. **39**, 695–698 (2010)
4. Cao, J.: Challenges in big data application development. In: 2013 Cross-strait Conference on Advanced Information Technology, pp. 5–22. Tamkang University (2013)
5. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H.: Big Data: The Next Frontier for Innovation, Competition, and Productivity. McKinsey Global Institute, New York (2011)
6. Linoff, G.S., Berry, M.J.: Data Mining Techniques: For Marketing Sales and Customer Relationship Management. Wiley, New York (2011)
7. Duhigg, C.: How Companies Learn Your Secrets. New York Times, New York (2012)

A Comparison of Feature-Combination for Example-Based Super Resolution

Shwu-Huey Yen(✉), Jen-Hui Tsao, and Wan-Ting Liao

PRIA. Laboratory, Department of Computer Science
and Information Engineering, Tamkang University, New Taipei,
Taiwan, Republic of China
105390@mail.tku.edu.tw, 700410268@s00.tku.edu.tw,
602410044@s02.tku.edu.tw

Abstract. Super resolution (SR) in computer vision is an important task. In this paper, we compared several common used features in image super resolution of example-based algorithms. To combine features, we develop a cascade framework to both solve the problem of deciding weights among features and to improve computation efficiency. Finally, we modify the framework to have an adaptive threshold such that not only the computation load is much reduced but the modified framework is suitable to any query image as well as various image databases.

Keywords: Super resolution (SR) · Example-based · Bicubic · Cascade

1 Introduction

Image super-resolution (SR) refers to the process by which a higher-resolution enhanced image is synthesized from one or more low-resolution images. Multiple-frame super resolution uses the sub-pixel shifts between multiple low resolution images of the same scene. The multi-frame SR problem was first addressed in [1], where they proposed a frequency domain approach. But subject to the sequential images are difficult to obtain in reality, the application of multiple-frame super resolution is not widely used. Single-frame SR methods use other parts of the low resolution images, or other unrelated images, to guess what the high-resolution image should look like. There are many single frame SR methods. Sun et al. [2] explored the gradient profile prior for local image structures and applied on SR. Such approaches are effective in preserving the edges in the zoomed image. Assuming that low resolution (LR) image patches and their high resolution (HR) counterparts share a similar geometry, Chang et al. [3] developed super resolution with neighbor embedding. One of the most famous is the example-based super resolution algorithm. As the name suggested, example-based method is to look for most suitable example from pre-prepared database to reconstruct the desired high resolution image. In general, the reconstructed result is database dependent. A database consisting of various examples will provide a better result but cause a burden in searching candidates. Glasner et al. [5] proposed a novel example-based method that does not rely on an external database. However, due to the limited database of examples from original image and its down-sample versions, the result

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 744–753, 2014.
DOI: 10.1007/978-3-319-13186-3_66

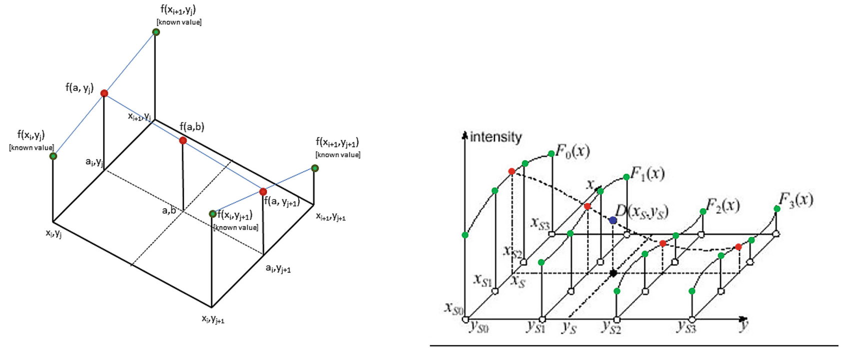
cannot be guaranteed. Presented by Freeman et al. proposed in 2002 [4], it divides a large amount of training images into small patches and uses them in the analytical process. Example-based super resolution is intuitive and simple to implement, the experimental results is also good for the visual, but because of the method proposed by Freeman learned target HR patch of size 5 × 5 from LR input patch of size 7 × 7, as opposed to the method proposed by Chang [3] learned target HR patch of size 12 × 12 from LR input patch of size 3 × 3, the method proposed by Freeman is inferior in terms of efficiency performance. Due to the attractive properties of intuitivism and simplicity, we explore the effectiveness of common used features of SR in the example-based method. In addition, based on the exploration conclusion, a cascade method of magnification factor 4 is proposed.

2 Related Work

Two methods most related to our study are bicubic interpolation and example based super resolution. We present a brief introduction on these methods in the following.

2.1 Bicubic Interpolation

Bicubic and bilinear interpolation are very common methods to resize images. The former is often chosen over the latter when speed is not an issue. In contrast to bilinear interpolation, which only takes 4 (2 × 2) pixels into account (Fig. 1a), bicubic interpolation considers 16 (4 × 4) pixels (Fig. 1b). Images resampled with bicubic interpolation are smoother and have fewer interpolation artifacts.



(a) Bilinear

(b) Bicubic(imagefrom[6])

Fig. 1. Interpolation methods

2.2 Example-Based Super Resolution

Example-based super-resolution algorithms involve a training set, which is usually composed of a large number of HR patches and their corresponding LR patches. The target input LR image is split into overlapping patches. Then, for each LR patch from the input image, one best-matched patch LR patches is selected from the training set. The corresponding HR patch is used to reconstruct the output HR image. Due to the fact that SR is to estimate missing high-resolution detail that is not present in the original LR image, Freeman et al. proposed an example-based method based on a Markov network [4]. The authors embedded two matching conditions into the network. One is that the LR patch from the training set should be similar to the input observed patch, while the other condition is that the contents of the corresponding HR patch should be consistent with its neighbors. They also proposed an one-pass algorithm to improve the performance.

3 Comparison on Various Features

To enlarge an image, the target LR image is divided into patches of size 3 × 3, and for each of these patches, we calculate three types of features: luminance (L), first order derivative (D) and bicubic intensity (B). Assume a 3 × 3 target LR patch, as in Fig. 2, features are described below where $I(p)$ is the intensity value of a point p.

- L: a 9-dim vector, $(I(a), I(b), \ldots, I(i))$
- D: a 18-dim vector, $(\partial I(a)/\partial x), \ldots, \partial I(i)/\partial x), \partial I(a)/\partial y), \ldots, \partial I(i)/\partial y)$
- B: a 144-dim vector of the bicubic result of the LR patch (magnification factor is 4)

a	*b*	*c*
d	*e*	*f*
g	*h*	*i*

Fig. 2. A target LR patch of 3×3

To prepare the database, a collection of sample images of various content are divided into 12 × 12 patches. Figure 3 shows the sample images used in the database (for all the experiments in the paper). To form a patch pair, for each 12 × 12 patch, we pair it with its down-sampled patch of size 3 × 3.

For a target LR image, we first use the Sobel edge detector to locate edge pixels. For every edge pixel P, we take a 3 × 3 patch centered at P so called the target patch. Three types of features (L, D, and B) are evaluated on the LR patches of the database (so called candidate LR patch) and the target patch as well. To find the most similar patch, we compute Euclidean distance of the features from between the target patch and candidate LR patches. Once such patch is found, its counterpart HR patch is adopted to reconstruct the HR version of the target image. As for the overlapped portion, we simply average the pixels. Finally, the metrics SSIM and PSNR are used for evaluation. To testify the effectiveness of features, we perform a series of tests. Figure 4 shows three test images (*Butterfly*, *Girl*, and *Couple*). In most of tests, for time efficiency, only partial images are used as shown on the right of the images.

3.1 Using Only One Feature

To find the most similar candidate patch, we first use only one feature. Table 1 shows the SR results using only single feature on three test images. As observed, feature B performs the best among features. However, it is computation intensive due to the large dimensions.

Table 1. Comparison on single feature

×4	Baby		Butterfly		Couple		Girl	
	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
Feature L (dim = 9)	0.63	24.2	0.54	17.62	0.77	25.44	0.73	27.86
Feature D (dim = 18)	0.62	24.32	0.52	17.19	0.79	26.17	0.74	28.27
Feature B (dim = 144)	0.65	25.04	0.52	16.68	0.8	26.13	0.74	28.02
Bicubic	**0.75**	**27.57**	**0.64**	**18.12**	**0.91**	**30.4**	**0.8**	**28.92**

3.2 Multiple Features in a Cascade Framework

Before multiple features comparison, we need to decide the way to combine different features. Features usually are linear combined with different weights. How to decide these weights to optimize the performance is a difficult task. A cascade framework can mitigate this problem. We divide the SR process into two or three rounds and each round we only use one feature. For example, if two features are used (i.e., 2 rounds), there will be only top k candidates from first round are kept for further computation for the second feature. By this way, although every patch pair in the database has to be compared with the target patch in the first round, there are only k comparisons required at the second round to find the best HR patch. Similarly, if three features are used, k_1 candidates are kept after the first round, and k_2 candidates are kept from those k_1 candidates after the second round, and finally, in the third round, the best candidate among k_2 candidates is used for SR reconstruction.

With this cascade framework, there are two new problems introduced: the order among these features and parameters k, or k_1 and k_2. For these problems, we design a complete test to find out the appropriate feature order and parameters.

The tests are first on combination of two features. As shown in Table 2, there are six possible combinations. For every target patch, “Feature_1-Feature_2” means that there are k best candidates kept according to the Feature_1 distance between the target patch and every candidate patch of the database. Then, the best candidate according to the Feature_2 distance between the target patch and those k candidates is used for SR.

To decide k, we first set k to be 250. As in Table 2, the combinations “L-D” and “B-D” outperform the other 4 combinations in *Butterfly* and *Couple*. Once tests completed, on the best feature combinations “L-D” and “B-D”, we again test on

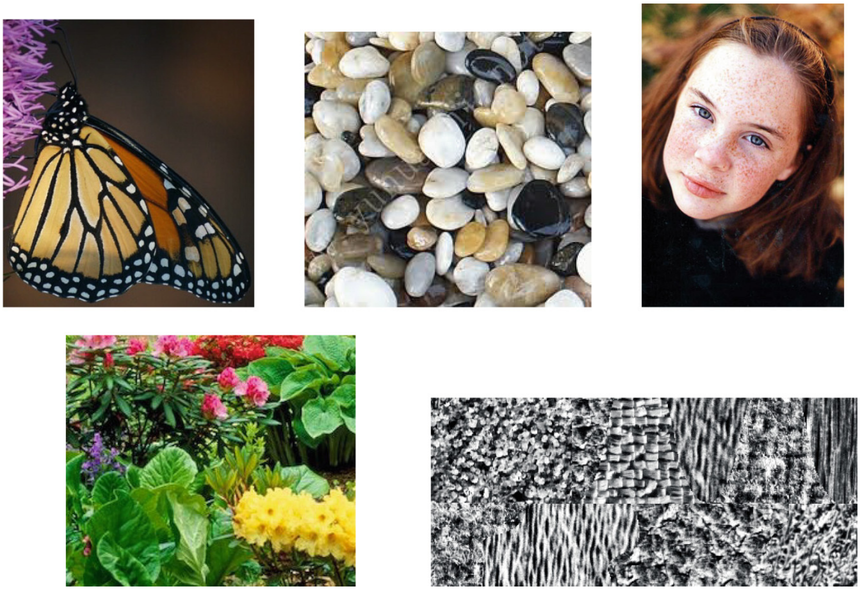
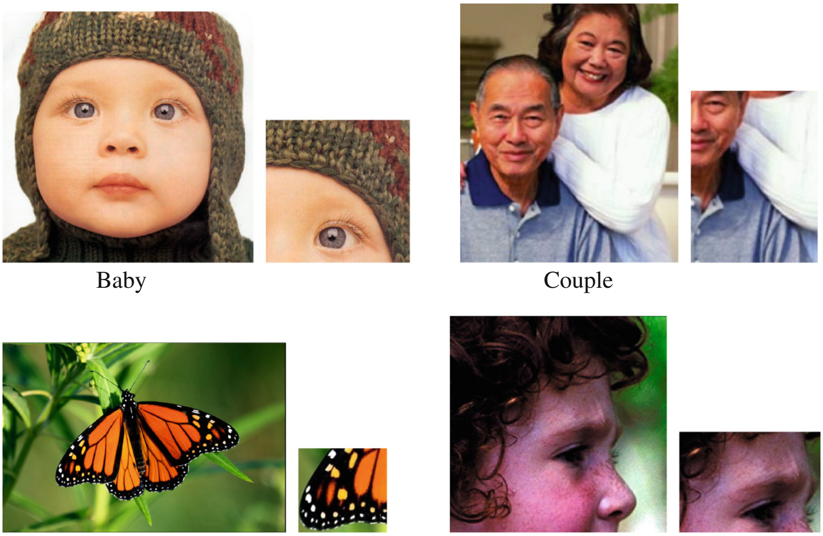


Fig. 3. Training images



Baby

Couple

Butterfly

Girl

Fig. 4. Testing images

Table 2. Two features combinations on k = 250

$\text{Feature}_1\text{-Feature}_2$	Butterfly		Couple	
	SSIM	PSNR	SSIM	PSNR
B-D	0.54	17.41	0.79	26.4
B-L	0.55	17.76	0.78	25.56
D-B	0.53	16.82	0.8	26.25
D-L	0.54	17.54	0.78	25.91
L-B	0.53	16.87	0.80	26.25
L-D	0.54	17.43	0.79	26.43
Bicubic	**0.64**	**18.12**	**0.91**	**30.40**

Table 3. Different k values on features "L-D" and "B-D"

Feature	K	Butterfly		Couple	
		SSIM	PSNR	SSIM	PSNR
L-D	50	0.55	17.56	0.80	26.56
	150	0.54	17.47	0.80	26.37
	250	0.54	17.43	0.79	26.43
	350	0.54	17.37	0.79	26.47
	450	0.53	17.32	0.79	26.41
B-D	50	0.53	17.32	0.80	26.71
	150	0.54	17.44	0.80	26.58
	250	0.54	17.41	0.79	26.40
	350	0.54	17.36	0.80	26.43
	450	0.54	17.36	0.79	26.48
Bicubic		**0.64**	**18.12**	**0.91**	**30.40**

different k values. The results are shown in Table 3. In general, when k = 50 or 150 it has better SSIM or better PSNR. In considering computation cost, 50 is a better choice.

In testing three features, we have done many possible combinations in feature ordering as well as parameters. Table 4 shows some of the results on three features combinations where "$\text{Feature}_1\text{-Feature}_2\text{-Feature}_3$" indicates the order of features and k_i is the number of best candidates kept after the *i*th round. The results in Table 4 are based on the best combination "L-D" and "B-D" of Table 2. To simplify the possible combinations, k_2 is set to be a dependent variable on k_1. According to the result of Table 3, we consider some possible values of k_1 ranging from 25 to 150. Overall, from Table 4, "L-B-D" with $k_1 = 20$, $k_2 = 5$ has the best performance.

Table 4. Three features combinations

Feature	K_1	K_2	Butterfly		Couple	
			SSIM	PSNR	SSIM	PSNR
L-B-D	25	0.25 * K_1	0.51	16.87	0.75	25.07
	50		0.45	16.35	0.76	25.13
	100		0.42	16.01	0.73	24.4
	150		0.41	15.74	0.71	24.18
	25	0.50 * K_1	0.49	16.62	0.75	24.96
	50		0.45	16.35	0.75	24.74
	100		0.43	16.22	0.72	23.94
	150		0.41	15.64	0.72	24.39
B-L-D	25	0.25 * K_1	0.49	16.53	0.77	25.44
	50		0.46	16.19	0.76	25.42
	100		0.44	16.19	0.74	24.68
	150		0.41	15.84	0.72	24.02
	25	0.50 * K_1	0.46	16.26	0.77	25.54
	50		0.44	15.96	0.74	24.91
	100		0.42	15.94	0.73	24.27
	150		0.41	15.86	0.72	24.05
Bicubic			**0.64**	**18.12**	**0.91**	**30.4**

4 Adaptive Thresholds

In the previous tests, we fixed the number of candidates kept for further examination. However, they may not be suitable if different database or different target LR image is used. To solve this problem, we propose an adaptive threshold T_i so that only those patches whose feature distance is less than T_i will be kept after round i. T_i is defined as

$$T_i = \min_i + n_i \cdot \sigma_i$$

with

$$n_i = \mathrm{MIN}(1, 0.5 \cdot (\mu_i - \mathrm{m}_i)/\sigma_i),$$

where m_i is the minimum feature distance when comparing to the i^{th} feature (i.e., the i^{th} round), μ_i and σ_i are the mean and standard deviation of all feature distances from the candidates for $i = 1, 2$. In particular, at round 1, the candidates are all the patches of the database; at round 2, the candidates are patches with feature distances less than T_1; finally, at round 3, the candidates are those from last round and their feature distances are less than T_2. We apply this adaptive threshold on feature "L-B-D" since it has a good performance according to Table 4.

In order to give privileges to those patches that have small feature distances in the previous round, the previous distance will be carried over to the distance in the current round. To do so, the distance has to be normalized since they have different

Table 5. Comparison of Adaptive and Fixed (L-D, K = 50)

	Adaptive/Fixed K(50)/Bicubic		
Picture	SSIM	PSNR	Time (s)
Baby	0.74/0.74/0.83	26.58/26.60/29.92	182/210/1
Butterfly	0.80/0.80/0.88	23.10/23.12/25.09	303/323/1
Couple	0.80/0.80/0.90	26.65/26.81/29.48	135/151/1
Girl	0.81/0.81/0.87	30.84/30.80/32.51	116/136/1

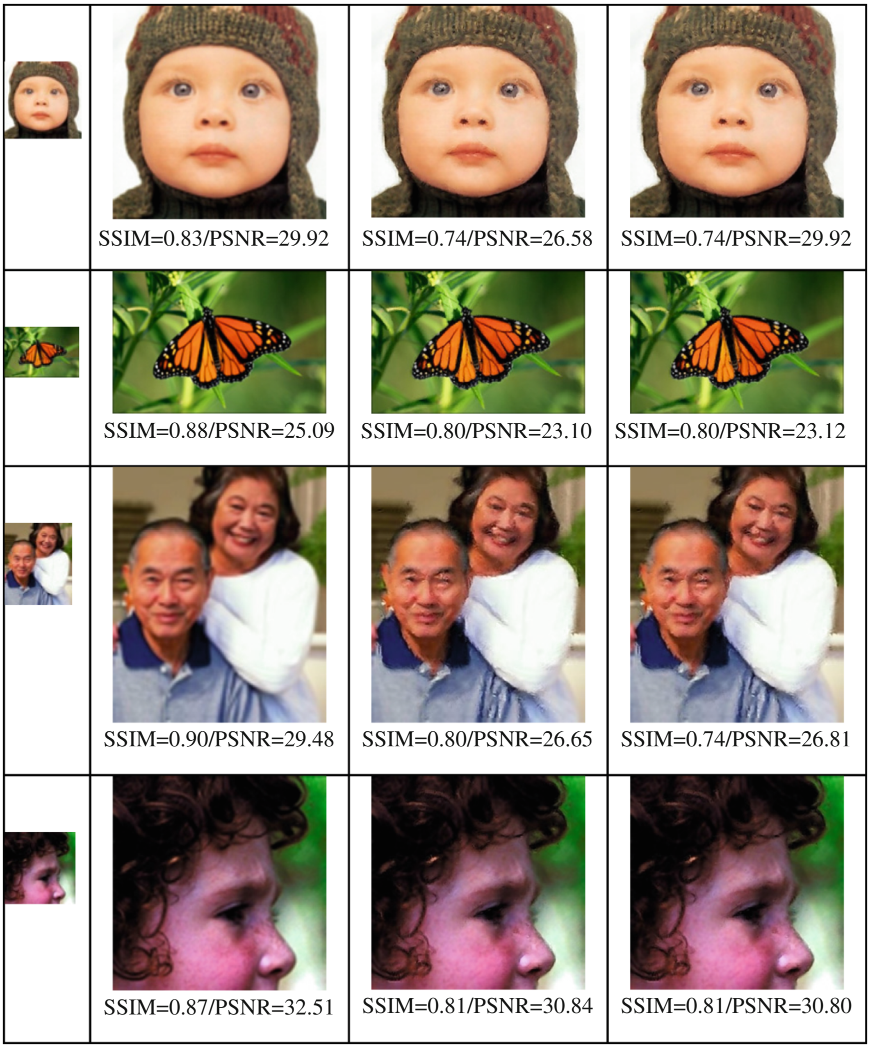


Fig. 5. Experimental results

dimensions. We first normalize the distance to be within 1 on each dimension. For a 3 × 3 candidate patch, we use notation $\mathrm{dist_i}$ to represent its feature distance comparing to target's after round *i*. Then, before round *i*, the initial distance for each candidate patch is defines as

$$\begin{aligned} i = 1 &: \ \mathrm{dist}_1 \leftarrow 0, \\ i = 2 &: \ \mathrm{dist}_2 \leftarrow \alpha \cdot [\mathrm{dist}_1/\mathrm{T\_Dist}_1] \cdot (114/9), \\ i = 3 &: \ \mathrm{dist}_3 \leftarrow \alpha \cdot [\mathrm{dist}_2/\mathrm{T\_Dist}_2] \cdot (18/144), \end{aligned}$$

where T_ $\mathrm{Dist_i}$ is the sum of all distances in round *i*. When carrying dist_1 to round two (feature one is L of dim = 9 and feature 2 is B of dim = 144), to be compatible to dist_2, a factor of (114/9) is multiplied in the initialization of dist_2. Similarly, feature 3 is D of dim = 18, a factor of (18/144) is multiplied in the initialization of dist_3.

In Table 5, statistics of SR results from bicubic, fixed (L-B-D, $k_1 = 20$, $k_2 = 5$), and adaptive (L-B-D) are shown. Some of results are given on Fig. 5. As observed, although values in SSIM and PSNR are not as good as bicubic, the reconstructed images of our both methods are visually pleasing and sharper. In comparing "fixed" and "adaptive" methods, they have similar performances but "adaptive" one can suit for different databases.

5 Conclusion and Future Work

In this paper, we explored three common used features in example-based super resolution algorithms and we also developed a cascade framework to solve the weighting problem in feature combination. We also provided an adaptive way in deciding the number of candidates for further checking in a cascade method. By this way, the computation burden in example-based method is much reduced and our method can suit for different database as well. In addition, we utilize the idea of distance initialization to give privileges to those better candidates in successive comparisons.

Because of the rich variability of images, a larger up-sample factor would make the HR patch harder to predict. In the future work, in order to enhance the accuracy of prediction, we will study how to reduce the size of the input patch to make the correct HR patch easier to predict. On the other hand, because example based super resolution methods are susceptible of training images, the study of diversity and availability of training images is also our future concern.

References

1. Huang, T.S., Tsai, R.Y.: Multi-frame image restoration and registration. Adv. Comput. Vis. Image Process. **1**, 317–339 (1984)
2. Sun, J., Xu, Z., Shum, H.-Y.: Steinke: image super-resolution using gradient profile prior. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8, 23–28 June 2008

3. Chang, H., Yeung, D.Y., Xiong, Y.: Super-resolution through neighbor embedding. In: Proceedings of the IEEE Computer Society of Conference on Computer Vision Pattern Recogonition, pp. 275–282 (2004)
4. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-based super-resolution. Comp. Graph. Appl. **22**(2), 56–65 (2002)
5. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: Proceedings of ICCV (2009), pp. 349–356 (2009)
6. http://software.intel.com/sites/products/documentation/hpc/ipp/ippi/ippi_appendices/Images/ippi_appB_2.jpg. Downloaded on June 2013

Comparative Analysis of 3D-Culture System for Murine Neonatal Heart Regeneration: A Systematic Approach for Big Gene Expression Data

Julia Tzu-Ya Weng[1](✉), Yi-Cheng Chen[2], Pei-Chann Chang[3],
Shin-Ping Huang[1], and Yu-Wei Chiu[4]

[1] Department of Computer Science and Engineering,
Yuan Ze University, Taoyuan, Taiwan
julweng@saturn.yzu.edu.tw, s1026015@mail.yzu.edu.tw
[2] Department of Computer Science and Information Engineering,
Tamkang University, New Taipei City, Tamsui, Taiwan
ycchen@mail.tku.edu.tw
[3] Department of Information Management, Yuan Ze University,
Taoyuan, Taiwan
iepchang@saturn.yzu.edu.tw
[4] Division of Cardiology Department of Internal Medicine,
Far-Eastern Memorial Hospital, Taipei, Taiwan
dtmed005@yahoo.com.tw

Abstract. Cardiovascular diseases are the leading cause of death worldwide. Loss or dysfunction of cardiomyocytes is associated with many forms of heart disease. The adult mammalian heart has a limited regenerative ability after damage, leading to the formation of fibrotic scar tissues, hypertrophy, contractile dysfunction and ultimately, organ failure. In contrast, neonatal mammalian cardiomyocytes retain a significant replenishing potential briefly after birth. There is increasing enthusiasm to grow neonatal cardiomyocytes in 3D culture systems to artificially restore heart function. Various scaffolds and matrices are available, but the molecular and cellular mechanisms underlying proliferation and differentiation of neonatal mammalian cardiomyocytes are not very well understood. Here, we utilize a systematic strategy to analyze the extensive genome-scale gene expression profiles of two different 3D constructs. We present a comprehensive comparison that may help improve the protocols for growing cardiomyocytes in a 3D culture system.

Keywords: Cardiomyocytes · Regeneration · 3-D culture system · Gene expression · Bioinformatics

1 Introduction

Cardiovascular diseases (CVDs) are the number one cause of death throughout the world [1], with the loss or dysfunction of cardiomyocytes being the key event leading to the formation of fibrotic scar tissues, pathological hypertrophy, contractile

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 754–764, 2014.
DOI: 10.1007/978-3-319-13186-3_67

dysfunction, and eventually resulting in heart failure [2]. In 2008, CVDs already account for 30 % of the total global deaths [1]. It is estimated that by 2030, more than 23 million people worldwide would die from CVDs [3].

Unfortunately, current pharmacological or surgical therapies for CVDs can slow the disease progression, but are unable to fully ameliorate cardiomyocyte loss or dysfunction [4]. Despite improvements in treatment, patients are still susceptible to an increased risk of chronic cardiac failure as scarring develops after injury [5]. The high disease burden, both from the perspectives of the healthcare system and the patients' quality of life, demands new therapeutic strategies to treat CVDs.

Advances in stem cell research and cell biology techniques have provided a wide variety of approaches to regenerate the heart from progenitor and cardiomyocytes derived from embryonic, neonatal, and adult cells. Though adult cardiomyocytes have been shown to have regenerative ability after injury, proliferation and differentiation occur at a much lower rate compared to embryonic and neonatal cells [6]. It is still unclear how and whether efficient cardiac regeneration can be stimulated and regulated in the adult mammalian heart.

In contrast, significant cardiac regeneration has been observed in the neonatal mammalian heart. In the mouse heart, surgical amputation of the ventricular apex has been shown to initiate cardiomyocyte proliferation, suggesting that the neonatal heart is capable of endogenous regenerative response [7]. However, such event only sustains for a brief period after birth. The cellular and molecular underpinnings of postnatal cell cycle arrest in cardiomyocytes and loss of cardiac regenerative capacity in the mammalian heart still remain elusive. Enhanced understanding of these key molecular events may offer novel insights for the development of ways to manipulate or extend the neonatal cardiac regenerative potential into adulthood.

In regenerative cell biology research, 3D cultures are gaining attention since they allow for the generation of 3D architecture that resembles more closely to the native environment of the tissue. Indeed, cells cultured in 3D and 2D conditions have been shown to differ in gene expression profile and phenotype [8]. Important markers of development and response to hormonal stimulation are more readily observed in 3D systems. 3D cultures have also been found to activate cell proliferation more efficiently compared to 2D designs [9]. It appears that 2D cultures may not be as flexible for manipulations or modulations, and thus, lack translational potential.

Owing to its increasing popularity in regenerative medical research, 3D constructs are now available with variable physical, chemical, and biological properties that can be manipulated to suit different purposes. For example, extracellular matrix prepared from decellularized cardiac tissue could be fabricated into 3D scaffolds and matrices, and injected to treat postmyocardial infarction injury [10–12]. Hydrogels and synthetic biodegradable materials can also be fabricated to create customized patterns and constructs [13, 14]. In fact, hydrogels have been combined with cell agents to successfully improve repair of cardiac function post injury [15–18].

Similar to the difference between 3D and 2D culture systems, different properties of 3D constructs may result in varying levels of success in cardiac tissue regeneration. We have compared among two 3D systems, PuraMatrix$^{\text{TM}}$ (BD Bioscience, U.S.A.) and Go-Matrix (Bio-Byblos Biomedical Co., Ltd., Taiwan), and the traditional 2D-cell culture system. The PuraMatrix$^{\text{TM}}$ construct is composed of peptide hydrogel with irregular pore arrangements, whereas Go-Matrix is made of gelatin from porcine skin with regular pore patterns. Compared to the more normal looking 2D-culture system and Go-Matrix, we observe irregular formation of muscle fibers that resembles myocardial fibrosis, as well as abnormal beating of the cardiac tissue manufactured from PuraMatrix^{TM} (unpublished results).

In the present study, we set to compare the gene expression differences between PuraMatrix^{TM} and Go-Matrix with microarray technology, using the 2D architecture as a baseline reference. The reason for including the 2D culture is that we have repeatedly and successfully generated cardiac tissues from this system. The next step in our research is to construct a 3D model that holds much more promises for clinical applications. Therefore, the 2D construct serves as a good normalization control for our analysis.

Microarray gene expression profiles comparing three different cell culture models, with each probe on the bio-chip repeatedly targeting (at least 10 times) each of the approximately 25,000 transcripts in the mouse genome. Such profiles represent a big dataset with great complexity and require a systematic pipeline combining the available bioinformatics tools to extract important biological meanings.

Here, we present an integrative system flow involving various tools for the analysis of gene expression differences between PuraMatrix^{TM} and Go-Matrix. These resources are integrated on the WEB-based Gene SeT AnaLysis Toolkit (WebGestalt) [19] and mirWalk [20], and include curated information from Gene Ontology (GO) [21], Kyoto Encyclopedia of Genes and Genomes (KEGG) [22], Pathway Commons [23], Wiki-Pathway [24], miRBase [25], Diana-microT [26], miRanda [27], miRDB [28], PicTar [29], PITA [30], RNA22 [31], and TargetScan [32]. We believe a combinatorial analysis utilizing publicly available bioinformatics tools should provide a better overview of the significance in the experiment design. Identifying the biological significance underlying the molecular differences between the two different 3D cell culture systems may help facilitate improvements in protocols for mending the “broken” heart.

2 Methods and Materials

2.1 System Flow

The system flow of our study is depicted in Fig. 1. Gene expression microarray experiment is performed to identify differentially expressed transcripts among two 3D cell culture systems, PuraMatrix^{TM} and Go-Matrix, with the traditional 2D culture as a normalization control. Differentially expressed were categorized into up- and down-regulated genes, and divided into PuraMatrix- and Go-Matrix-specific groups. The differentially expressed genes are subsequently subjected to bioinformatics analysis WebGestalt [19] and mirWalk [20].

2.2 Cardiomyocyte Isolation

All procedures were approved by the Institutional Animal Care and Use Committee of the Far-Eastern Memorial Hospital. Neonatal B6 mice were sacrificed by de-capitation

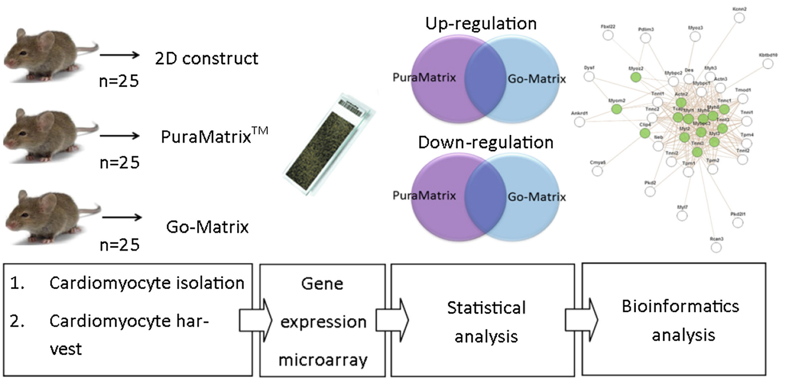


Fig. 1. System flow of our wet bench and bioinformatics analyses.

on postnatal day 3. 25 neonatal mice underwent the surgical procedures for cardiomyocyte isolation. The hearts were quickly excised and transferred into ice-cold phosphate buffer saline (PBS, Sigma), and quickly minced in a solution of colla-genase A + B solution (10 mg/ml in PBS). Cells in the supernatants were resuspended in claycomb medium with 10 % fetal bovine serum (FBS, Gibco) and plated for 1.5 h at 37 °C. The attached fibroblasts and endothelial cells were removed and the cardiomyocytes were isolated in the suspension solution.

2.3 Cell Seeding of Constructs

Cardiomyocytes were cultured on gelatin/fibronectin-coated tissue-culture flasks in claycomb medium at 37 °C in a CO_2 incubator. Cells harvested and cultured in this method served as the 2D control group. To prepare PuraMatrix™, 1 % stock solution was mixed with 20 % sucrose (Sigma–Aldrich, Sweden AB) and DMEM containing Ca^{2+} and Mg^{2+} in a 2:1:1 ratio, giving 0.5 % hydrogel. Cardiomyocytes were detached from the culture flasks with trypsin/EDTA, concentrated in claycomb medium and immersed with PuraMatrix™ peptide hydrogel. Go-Matrix is a gelatin scaffold fixed on tissue-culture flasks. The claycomb medium containing cardiomyocytes was dropped onto the dry Go Matrix. In total, 106 cells were cultured in each group.

2.4 RNA Isolation

RNA was isolated from harvested cardiomyocyte cells. RNA quality was determined by an OD 260/280 ratio ≥ 1.8, and OD 260/230 ratio ≥ 1.5 on a spectrophotometer and by the intensity of the 18S and 28S rRNA bands on a 1 % formaldehydeagarose gel. RNA quantity was detected by a spectrophotometer. RNA integrity was examined on an Agilent Bioanalyzer. RNA with a 2100 RIN (RNA integrity number) ≥ 6.0 and 28S/18S > 0.7 was subjected to microarray analysis.

2.5 Gene Expression Analysis

Total RNA samples from 25 mice in each construct are pooled and subsequently subjected to SurePrint G3 Mouse GE 8x60 K Microarray (Agilent Technologies, U.S.A.). Data are analyzed using R/Bioconductor. PuraMatrix$^{\text{TM}}$ and Go-Matrix expression profiles were normalized against that of the cardiomyocytes prepared from the traditional 2D cell culture system. Genes showing significant differential expression between normalized PuraMatrixTM and Go-Matrix (absolute value of $\log_2$ ratio > 1.0, FDR < 0.05) were categorized into PuraMatrix$^{\text{TM}}$ -specific down- and up-regulated genes, and Go-Matrix-specific down- and up-regulated genes.

2.6 Bioinformatics Analysis

A flow chart describing our bioinformatics analysis is given in Fig. 2. Differentially expressed genes were used as input for the bioinformatics analysis of gene ontology (GO) enrichment analysis and pathway enrichment analysis. Multiple testing bias is adjusted by a Bonferroni threshold of $p < 0.05$. These analyses are performed on the open analytical platform in WebGestalt [19], which integrates GO [21], KEGG [22], Pathway Commons [23], and WikiPathways [24]. microRNA target prediction analysis is conducted in mirWalk [20], which, in addition to the developers' own analytical algorithms, also makes comparisons with other microRNA databases, including miRBase [25], Diana-microT [26], miRanda [27], miRDB [28], PicTar [29], PITA [30], RNA22 [31], and TargetScan [32].

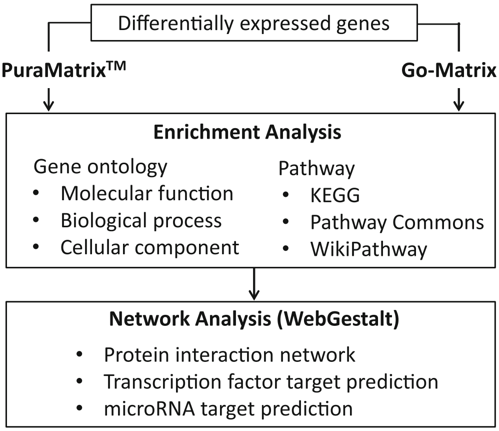


Fig. 2. Bioinformatics analysis workflow

3 Results and Discussions

In total, there are 516 up-regulated and 338 down-regulated genes, and 401 up-regulated and 292 down-regulated genes specific to the PuraMatrix$^{\text{TM}}$ and Go-Matrix expression profiles, respectively (Fig. 3). It appears that the Go-Matrix expression profile may be more similar to the traditional 2D culture system. For both 3D constructs, the top three most enriched biological processes are: biological regulation, metabolic process, and response to stimulus. The top three most important cellular components include the membrane, nucleus, and macromolecular complex. For molecular functions, both expression profiles are enriched in protein binding, ion binding, and metabolite binding.

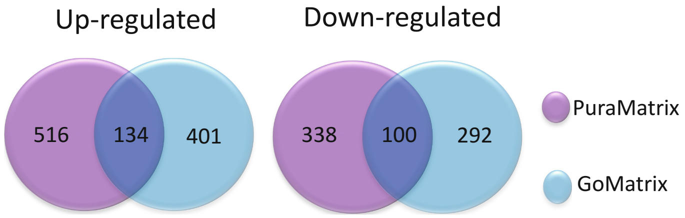


Fig. 3. Number of Differentially expressed genes from the PuraMatrix^{TM} and Go-Matrix expression profiles

PuraMatrix^{TM} and Go-Matrix expression profiles appear to be similar in muscle contraction pathways, though the gene expression patterns are different. In particular, the striated muscle contraction pathway that is enriched in both PuraMatrix^{TM} and Go-Matrix gene list. However, the two 3D cell culture systems differ in the genes that are differentially expressed in this pathway.

Though similar in some pathways, PuraMatrix^{TM} and Go-Matrix expression profiles appear to be enriched in different types of cellular pathways. For instance, most of the differentially expressed genes in the PuraMatrix^{TM} construct seem to belong to muscle contraction and cell adhesion related pathways (Table 1). In contrast, Go-Matrix-specific differentially expressed genes are enriched in pathways involving membrane receptor interactions.

Cell culture observations indicate that the PuraMatrix^{TM} cell culture system tends to generate tissue with abnormal muscle fiber arrangements and contractions (unpublished results). This is consistent with our gene expression profile analysis in that most of the differentially expressed genes in the PuraMatrix^{TM} design belong to cardiomyopathy pathways. For example, among the list of PuraMatrix^{TM}-specific up-regulated genes involved in dilated cardiomyopathy, the ryanodine receptor 2 (*Ryr2*) gene, when silenced by RNAi, protects rat cardiomyocytes from simulated ischemia-induced injury [33]. The *Pln* (phospholabam) gene, when deleted or suppressed, is known to restore normal cardiomyocyte contraction and slow the progression towards cardiomyopathy [34]. This suggests that the observed phenotype and gene expression changes associated with the PuraMatrix^{TM} construct of cardiomyoctes may serve as a potential basis for the future development of a cardiomyopathy related disease model.

In contrast, the differentially expressed genes in the Go-matrix seem to be related to the development of cardiac cells and the establishment of cell-cell contact (Table 1). This corresponds to the ordered arrangements of muscle fibers and regular muscle contractions of the Go-Matrix cardiomyocyte culture (unpublished results). The enrichment analysis result is also consistent with similar findings in fetal sheep, in which neuroactive-ligand-receptor interaction, cytokine-cytokine receptor interaction,

Table 1. List of top 3 most enriched pathways in PuraMatrix^{TM} and Go-Matrix expression profiles as identified by KEGG and WikiPathway.

PuraMatrix^{TM} enriched pathways	Go-Matrix enriched pathways
KEGG (up-regulation)	
Dilated cardiomyopathy	Neuroactive-ligand-receptor interaction
Hypertrophic cardiomyopathy	Cytokine-cytokine receptor interaction
Pathways in cancer	Complement and coagulation cascades
KEGG (down-regulation)	
Vascular smooth muscle contraction	Axon guidance
Neuroactive-ligand-receptor interaction	Notch signaling pathway
Protein digestion and absorption	Melanogenesis
WikiPathway (up-regulation)	
Striated muscle contraction	Complement and coagulation cascades
Focal adhesion	Non-odorant G-protein coupled receptors
PPAR signaling pathway	Striated muscle contraction
WikiPathway (down-regulation)	
Striated muscle contraction	Delta-Notch signaling pathway
Integrin-mediated cell adhesion	Cholesterol biosynthesis
Focal adhesion	Complement and coagulation cascades

complement and coagulation cascades, etc. were associated cardiogenesis [35]. In particular, the Notch signaling pathway has been implicated in development of cardiac neural crest cells and heart valves [36].

Interestingly, differentially expressed genes in both PuraMatrix^{TM} and Go-matrix profiles are enriched in the striated muscle contraction pathway, though these genes play differing roles in this particular process (Fig. 4). This indicates that differences in the modulation of gene expression have specific effect on the muscle contraction phenotype. Such differences may contribute to the phenotypic variations between the cardiomyocytes grown in these two culture systems.

The interactions among the differentially expressed genes involved in the striated muscle contraction pathway may reveal the molecular underpinnings of the differences in muscle contraction between cardiomyoctyes constructed by PuraMatrixTM and Go-Matrix. Protein-interaction and microRNA target analysis suggest that three genes would be strong potential candidates for further functional study of cardiac development. These genes are *Myh6* (myosin, heavy polypeptide 6, cardiac muscle, alpha), which is expressed in the atria and associated with congenital heart defects [37]; *Tnnc1* (troponin C type 1), which plays a regulatory role in the beating of the early developing heart [38]; *Des* (desmin), which is an intermediate filament gene important for muscle architecture [39].

Our results suggest that that *Myh6*, *Tnnc1*, and *Des* are likely interacting with each other, as well as with other important molecular signatures of the cardiac contraction pathway (Fig. 5). Furthermore, their expression may be regulated by several microRNAs that have been implicated to play important roles in cardiac function. For instance, the mir-29 family is responsible for modulating the development of cardiac

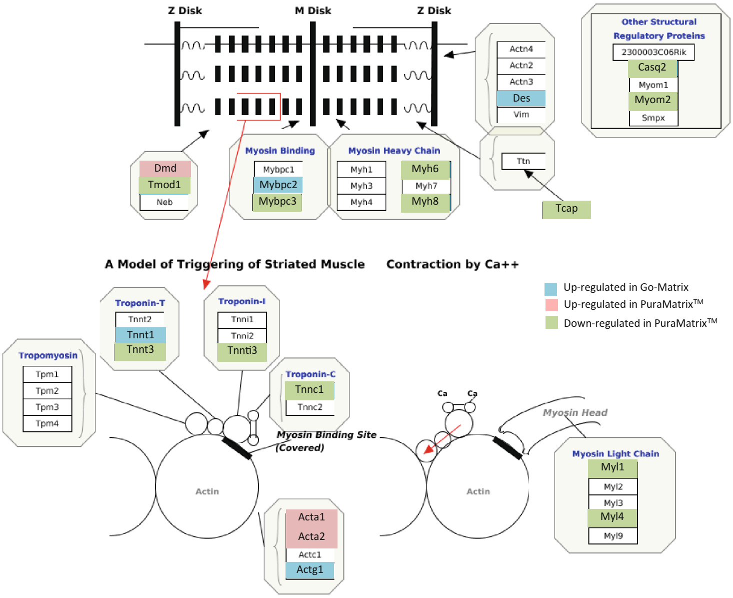


Fig. 4. PuraMatrix^{TM}-specific and Go-Matrix-specific differentially expressed genes involved in the striated muscle contraction pathway

fibrosis after injury [40]. In addition, mir-208a is known to be a regulator of cardiac stress response [41]. The interaction network presented in Fig. 5 may represent a potential modulatory mechanism underlying the differences the in muscle contraction phenotype between PuraMatrix^{TM} and Go-Matrix.

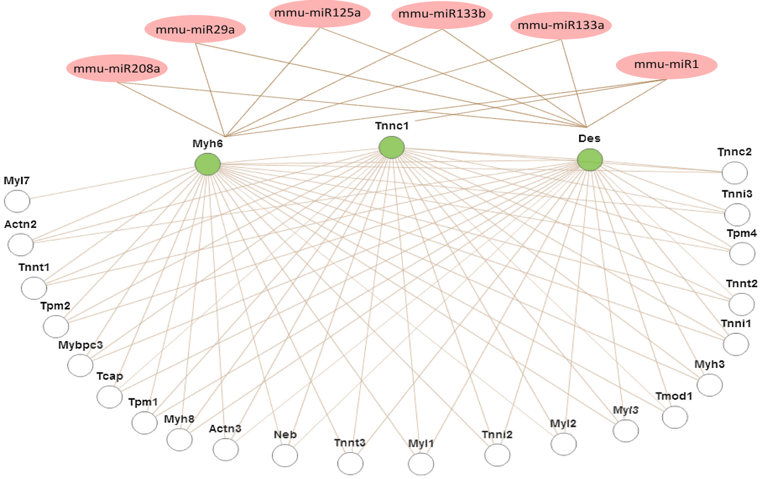


Fig. 5. Protein interaction network showing the interaction among Myh6, Tnnc1, Des with other genes and corresponding validated regulatory microRNAs from the striated muscle contraction pathway

4 Conclusion

3D cell culture systems are gaining attention in regenerative medical research. These constructs can be manipulated to suit different purposes, either for understanding development and disease mechanisms, or establishing a disease model. The ways in which these systems can be manipulated is of particular interest to regenerative medicine. Here, we present a systematic pipeline, integrating a wide variety of bioinformatics tools, to extract important biological meanings from the genome-scale differences between two 3D systems utilized for cardiomyocyte culturing. Our analysis presents a preliminary picture of the significant pathways and interactions associated with cardiomyocyte development in 3D. Our findings may have special implications for the future establishment of a better model for murine neonatal heart regeneration.

References

1. Organization, W.H.: Global status report on noncommunicable diseases 2010. World Health Organization (2011)
2. Kajstura, J., Urbanek, K., Perl, S., Hosoda, T., Zheng, H., Ogorek, B., Ferreira-Martins, J., Goichberg, P., Rondon-Clavo, C., Sanada, F., D'Amario, D., Rota, M., Del Monte, F., Orlic, D., Tisdale, J., Leri, A., Anversa, P.: Cardiomyogenesis in the adult human heart. Circ. Res. **107**, 305–315 (2010)
3. Mathers, C.D., Loncar, D.: Projections of global mortality and burden of disease from 2002 to 2030. PLoS Med. **3**, e442 (2006)
4. Steinhauser, M.L., Lee, R.T.: Regeneration of the heart. EMBO Mol. Med. **3**, 701–712 (2011)
5. Bartunek, J., Behfar, A., Dolatabadi, D., Vanderheyden, M., Ostojic, M., Dens, J., El Nakadi, B., Banovic, M., Beleslin, B., Vrolix, M., Legrand, V., Vrints, C., Vanoverschelde, J.L., Crespo-Diaz, R., Homsy, C., Tendera, M., Waldman, S., Wijns, W., Terzic, A.: Cardiopoietic stem cell therapy in heart failure: the C-CURE (Cardiopoietic stem Cell therapy in heart failURE) multicenter randomized trial with lineage-specified biologics. J. Am. Coll. Cardiol. **61**, 2329–2338 (2013)
6. Hsieh, P.C.H., Segers, V.F.M., Davis, M.E., MacGillivray, C., Gannon, J., Molkentin, J.D., Robbins, J., Lee, R.T.: Evidence from a genetic fate-mapping study that stem cells refresh adult mammalian cardiomyocytes after injury. Nat. Med. **13**, 970–974 (2007)
7. Porrello, E.R., Mahmoud, A.I., Simpson, E., Hill, J.A., Richardson, J.A., Olson, E.N., Sadek, H.A.: Transient regenerative potential of the neonatal mouse heart. Science **331**, 1078–1080 (2011)
8. Akins Jr., R.E., Rockwood, D., Robinson, K.G., Sandusky, D., Rabolt, J., Pizarro, C.: Three-dimensional culture alters primary cardiac cell phenotype. Tissue Eng. Part A **16**, 629–641 (2010)
9. Kellar, R.S., Landeen, L.K., Shepherd, B.R., Naughton, G.K., Ratcliffe, A., Williams, S.K.: Scaffold-based three-dimensional human fibroblast culture provides a structural matrix that supports angiogenesis in infarcted heart tissue. Circulation **104**, 2063–2068 (2001)
10. Singelyn, J.M., Christman, K.L.: Injectable Materials for the Treatment of Myocardial Infarction and Heart Failure: The Promise of Decellularized Matrices. J. Cardiovasc. Transl. **3**, 478–486 (2010)

11. Godier-Furnemont, A.F.G., Martens, T.P., Koeckert, M.S., Wan, L., Parks, J., Arai, K., Zhang, G.P., Hudson, B., Homma, S., Vunjak-Novakovic, G.: Composite scaffold provides a cell delivery platform for cardiovascular repair. Proc. Natl. Acad. Sci. U.S.A. **108**, 7974–7979 (2011)
12. Singelyn, J.M., Sundaramurthy, P., Johnson, T.D., Schup-Magoffin, P.J., Hu, D.P., Faulk, D. M., Wang, J., Mayle, K.M., Bartels, K., Salvatore, M., Kinsey, A.M., Demaria, A.N., Dib, N., Christman, K.L.: Catheter-deliverable hydrogel derived from decellularized ventricular extracellular matrix increases endogenous cardiomyocytes and preserves cardiac function post-myocardial infarction. J. Am. Coll. Cardiol. **59**, 751–763 (2012)
13. Khademhosseini, A., Langer, R., Borenstein, J., Vacanti, J.P.: Microscale technologies for tissue engineering and biology. Proc. Natl. Acad. Sci. U.S.A **103**, 2480–2487 (2006)
14. Rane, A.A., Chuang, J.S., Shah, A., Hu, D.P., Dalton, N.D., Gu, Y., Peterson, K.L., Omens, J.H., Christman, K.L.: Increased infarct wall thickness by a bio-inert material is insufficient to prevent negative left ventricular remodeling after myocardial infarction. PLoS ONE **6**, e21571 (2011)
15. Garbern, J.C., Minami, E., Stayton, P.S., Murry, C.E.: Delivery of basic fibroblast growth factor with a pH-responsive, injectable hydrogel to improve angiogenesis in infarcted myocardium. Biomaterials **32**, 2407–2416 (2011)
16. Lu, W.N., Lu, S.H., Wang, H.B., Li, D.X., Duan, C.M., Liu, Z.Q., Hao, T., He, W.J., Xu, B., Fu, Q., Song, Y.C., Xie, X.H., Wang, C.Y.: Functional improvement of infarcted heart by co-injection of embryonic stem cells with temperature-responsive chitosan hydrogel. Tissue Eng. Part A **15**, 1437–1447 (2009)
17. Huang, N.F., Yu, J.S., Sievers, R., Li, S., Lee, R.J.: Injectable biopolymers enhance angiogenesis after myocardial infarction. Tissue Eng. **11**, 1860–1866 (2005)
18. Davis, M.E., Motion, J.P., Narmoneva, D.A., Takahashi, T., Hakuno, D., Kamm, R.D., Zhang, S., Lee, R.T.: Injectable self-assembling peptide nanofibers create intramyocardial microenvironments for endothelial cells. Circulation **111**, 442–450 (2005)
19. Wang, J., Duncan, D., Shi, Z., Zhang, B.: WEB-based GEne SeT AnaLysis Toolkit (WebGestalt): update 2013. Nucleic Acids Res. **41**, W77–W83 (2013)
20. Dweep, H., Sticht, C., Pandey, P., Gretz, N.: miRWalk–database: prediction of possible miRNA binding sites by "walking" the genes of three genomes. J. Biomed. Inform. **44**, 839–847 (2011)
21. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. **25**, 25–29 (2000)
22. Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T., Yamanishi, Y.: KEGG for linking genomes to life and the environment. Nucleic Acids Res. **36**, D480–D484 (2008)
23. Cerami, E.G., Gross, B.E., Demir, E., Rodchenkov, I., Babur, O., Anwar, N., Schultz, N., Bader, G.D., Sander, C.: Pathway Commons, a web resource for biological pathway data. Nucleic Acids Res. **39**, D685–D690 (2011)
24. Pico, A.R., Kelder, T., van Iersel, M.P., Hanspers, K., Conklin, B.R., Evelo, C.: WikiPathways: Pathway editing for the people. PLoS Biol. **6**, 1403–1407 (2008)
25. Griffiths-Jones, S., Saini, H.K., van Dongen, S., Enright, A.J.: miRBase: tools for microRNA genomics. Nucleic Acids Res. **36**, D154–D158 (2008)

26. Maragkakis, M., Reczko, M., Simossis, V.A., Alexiou, P., Papadopoulos, G.L., Dalamagas, T., Giannopoulos, G., Goumas, G., Koukis, E., Kourtis, K., Vergoulis, T., Koziris, N., Sellis, T., Tsanakas, P., Hatzigeorgiou, A.G.: DIANA-microT web server: elucidating microRNA functions through target prediction. Nucleic Acids Res. **37**, W273–W276 (2009)
27. Betel, D., Wilson, M., Gabow, A., Marks, D.S., Sander, C.: The microRNA.org resource: targets and expression. Nucleic Acids Res. **36**, D149–D153 (2008)
28. Wang, X.: miRDB: a microRNA target prediction and functional annotation database with a wiki interface. RNA **14**, 1012–1017 (2008)
29. Anders, G., Mackowiak, S.D., Jens, M., Maaskola, J., Kuntzagk, A., Rajewsky, N., Landthaler, M., Dieterich, C.: doRiNA: a database of RNA interactions in post-transcriptional regulation. Nucleic Acids Res. **40**, D180–D186 (2012)
30. Kertesz, M., Iovino, N., Unnerstall, U., Gaul, U., Segal, E.: The role of site accessibility in microRNA target recognition. Nat. Genet. **39**, 1278–1284 (2007)
31. Loher, P., Rigoutsos, I.: Interactive exploration of RNA22 microRNA target predictions. Bioinformatics **28**, 3322–3323 (2012)
32. Grimson, A., Farh, K.K.H., Johnston, W.K., Garrett-Engele, P., Lim, L.P., Bartel, D.P.: MicroRNA targeting specificity in mammals: Determinants beyond seed pairing. Mol. Cell **27**, 91–105 (2007)
33. Guo, Z., Wang, S., Jiao, Q., Xu, M., Gao, F.: RNAi targeting ryanodine receptor 2 protects rat cardiomyocytes from injury caused by simulated ischemia-reperfusion. Biomed. Pharmacotherapie **64**, 184–190 (2010)
34. Hoshijima, M., Ikeda, Y., Iwanaga, Y., Minamisawa, S., Date, M.O., Gu, Y., Iwatate, M., Li, M., Wang, L., Wilson, J.M., Wang, Y., Ross Jr., J., Chien, K.R.: Chronic suppression of heart-failure progression by a pseudophosphorylated mutant of phospholamban via in vivo cardiac rAAV gene delivery. Nat. Med. **8**, 864–871 (2002)
35. Cox, L.A., Glenn, J.P., Spradling, K.D., Nijland, M.J., Garcia, R., Nathanielsz, P.W., Ford, S.P.: A genome resource to address mechanisms of developmental programming: determination of the fetal sheep heart transcriptome. J. Physiol-London **590**, 2873–2884 (2012)
36. Niessen, K., Karsan, A.: Notch signaling in cardiac development. Circ. Res. **102**, 1169–1181 (2008)
37. Granados-Riveron, J.T., Ghosh, T.K., Pope, M., Bu'Lock, F., Thornborough, C., Eason, J., Kirk, E.P., Fatkin, D., Feneley, M.P., Harvey, R.P., Armour, J.A., David Brook, J.: Alpha-cardiac myosin heavy chain (MYH6) mutations affecting myofibril formation are associated with congenital heart defects. Hum. Mol. Genet. **19**, 4007–4016 (2010)
38. Stoutamyer, A., Dhoot, G.K.: Transient expression of fast troponin C transcripts in embryonic quail heart. J. Muscle Res. Cell Motil. **26**, 237–245 (2005)
39. Sam, M., Shah, S., Friden, J., Milner, D.J., Capetanaki, Y., Lieber, R.L.: Desmin knockout muscles generate lower stress and are less vulnerable to injury compared with wild-type muscles. Am. J. Physiol-Cell Ph. **279**, C1116–C1122 (2000)
40. van Rooij, E., Sutherland, L.B., Thatcher, J.E., DiMaio, J.M., Naseem, R.H., Marshall, W.S., Hill, J.A., Olson, E.N.: Dysregulation of microRNAs after myocardial infarction reveals a role of miR-29 in cardiac fibrosis. Proc. Natl. Acad. Sci. U.S.A. **105**, 13027–13032 (2008)
41. Callis, T.E., Pandya, K., Seok, H.Y., Tang, R.H., Tatsuguchi, M., Huang, Z.P., Chen, J.F., Deng, Z.L., Gunn, B., Shumate, J., Willis, M.S., Selzman, C.H., Wang, D.Z.: MicroRNA-208a is a regulator of cardiac hypertrophy and conduction in mice. J. Clin. Invest. **119**, 2772–2786 (2009)

Data Collection and Analysis from Social Network – Profile Analyzer System (PAS)

Wei-Hao Chang, Bing Li, and Xin Fang(✉)

SIGI Technologies Inc., Beijing, China
{oliver.jang, singh.fang}@gmail.com,
libing_sigi@outlook.com

Abstract. Today we have many practices with applying either big data or data mining in various of subjects. For example, many advisory companies use their methodologies as a consultant service to assist other companies to focus on their business. However, most of these practices are focus on marketing insights and public opinion monitoring but lack of quality of data sources. In the purpose of the development of Profile Analyzer System, we are more likely aimed at analyzing individual insights such as personal behavior, habits, personality, etc. among the mass social network data. In this paper, we introduce how we realize PAS in the usage of social network data sources and represent the corresponding analysis framework as the technique basis of this system. The profile characteristics is also visualized with a set of analyzed dataset in practice.

Keywords: Data collection · Social network · And profile analysis

1 Introduction

Social media with big data is a very hot topic in the recent years. However, most of the related applications are practiced in business modules [1]. Since there are many social network companies relatively open to the whole market [3]. As holding tons of user data, these companies are willing to share the data source for the purpose of being able to occupy a greater market share [2]. Therefore, we decided to build up a system that is valuable and much different from other systems since the analyzer system based on big data theory has become very popular and with limited differentiation.

Profile Analyzer System is a platform-liked tool to help people to understand themselves more easily. Even if this tool is used for business purpose, we may help the company make better in precision marketing or directional marketing. The rest of the article is organized as follows: first chapter is to introduce how we collection social media data from the websites. In next chapter, we will tell the data source we used and what data type is accepted in our system. In chapter three, data dimension is described as an important factor in the whole process. Finally, the process of analysis profile will be discussed in detail.

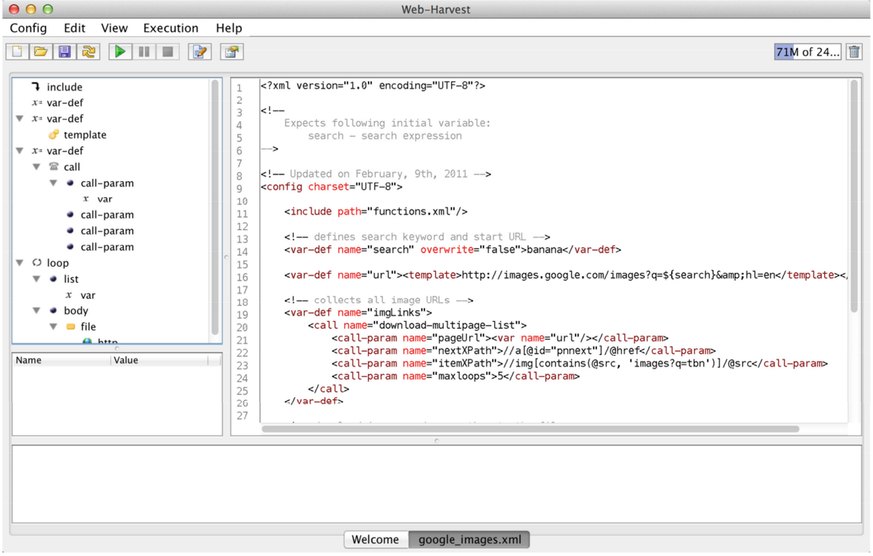
2 Data Crawler

Data collected in the era of data outbreak is a very important part, and the way to collect data differs the data in form. In-vehicle devices while on moving, we can collect

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 765–772, 2014.
DOI: 10.1007/978-3-319-13186-3_68

the coordinates of the vehicle, driving speed, and even driver's mental state [4, 5]. These data collection can tremendously generate valuable applications. It is possible to use the autopilot with navigation applications if the map information can be collected continuously. Moreover, the auxiliary parking pattern can be realized if the surrounding objects data can be transmitted to the vehicle. We may be able to analyze personal schedule in a whole day if we can collect the data from mobile device, even to calculate his/her working place or know when he/she will go to the gas station to fill up the vehicle. Selecting the data source is an important part that affects the application usage. Several well-known crawler tools for data mining mainly used in PAS; the following is the introduction of these tools:

WebHarvest (Java): Web harvest is focus on html/xml based websites. It can easily leverage well-proved xml and text-based html procession. PAS used this tool as a part of crawler tools in routine job (Fig. 1).



```
<?xml version="1.0" encoding="UTF-8"?>

<!--
    Expects following initial variable:
        search - search expression
-->

<!-- Updated on February, 9th, 2011 -->
<config charset="UTF-8">

    <include path="functions.xml"/>

    <!-- defines search keyword and start URL -->
    <var-def name="search" overwrite="false">banana</var-def>

    <var-def name="url"><template>http://images.google.com/images?q=${search}&amp;hl=en</template><

    <!-- collects all image URLs -->
    <var-def name="imgLinks">
        <call name="download-multipage-list">
            <call-param name="pageUrl"><var name="url"/></call-param>
            <call-param name="nextXPath">//a[@id="pnnext"]/@href</call-param>
            <call-param name="itemXPath">//img[contains(@src, 'images?q=tbn')]/@src</call-param>
            <call-param name="maxloops">5</call-param>
        </call>
    </var-def>
```

Fig. 1. Screenshot of WebHarvest

Crawler4j (Java): This tool is executed with apache ant which is a well-known java-based task manager within command-line interface. Ant tool controls the batch job in routine work and crawler4j is in charge of being executed for multi-thread tasks in use for crawling on multiple websites at once. In other word, crawler4j has a crucial feature that will automatically detect the non-utf8 webpage to deal with the character issues.

3 Data Type and Data Source

In general, the information stored in social networking sites mostly presented as: a short-text format and long-text format, picture, short-term video, or in the form of hyperlink existed. However, most technique in big data area has supported for multiform at storage such as Hadoop Distributed File System [6]. PAS takes advantage of many data sources of social media data, so as to consider extensive use of various information and data. In phase one, we would rather crawl the social data in the format of text file than the format of other types. We will follow up the data in the form of graphics or video to analyze the data to optimize the predictive analysis of efficiency in later phases.

After the data format clearly defined, we need to consider the forms and methodology of data storage because the data stored can affect the efficiency of data read back and calculations. There are two ways to store the data: ETL (Extract, Transform, and Load) and ELT (Extract, Load, and Transform). Because of the general business anaysis will be reused with crawling data, preserving the original data can reduce the probability of re-crawl data. While there are different business goal oriented, this approach would be suitable for using ELT form of data storage. Data are centrally stored in RAW data warehouse; the RAW data will be converted to business data set for analysis when you want to run your analyzer programs.

PAS system is aimed to analyze anonymous user profile. It will be more efficient to pre-convert RAW data into business data set because the data set can be only used for once in most cases. Analysis to calculate the time spent much faster than the way of ELT.

PAS system will go through an anonymous social-network user as main account, and extend to dig other related information in social networking websites on the basis of the main account. There are several major sources of social-network sites are as follows:

1. Social community: Sina weibo, Facebook, Google+
2. Short text media: Twitter.
3. Video media: YouTube, Youku.
4. E-Commerce media: Amazon, Taobao, Jingdong.

PAS will firstly expect the user's master account is the same identity used in other community. The first excavation, therefore, will grab the related data source derived from the main account. If not, PAS will try to search the data source from other sites via user behavior or personal information such as birth date, general location, or habits in comparison to the main account's information. In general, the master account was associated with the email account, account name, or similar-name account.

The second stage, we grab [8] dataset associated with the user's master account. These collected data are, for example, re-twitter counts, weibo posts, personal tags, or "like" events. We regard these dimensional dataset as our core variables that may tremendously affect our analysis process.

In order to deal with the connection between the user's master account and his/her related friends, we use XFN (Xhtml Friends Network) as one of our means to find master user's relationship. With the documents in terms of HTML and XHTML, it can be traced from the "rel" attribute on a hyperlink, which indicate the weibos belong to the main user's friends. The social network of the user was therefore linked up via the XFN values.

4 Dimension Efficiency

After we defined the data sources and data processing, we have to precise the direction of data source we crawl. Due to the traffic restriction of sites operation, it may not allow us to crawl the whole data uninterruptedly. Therefore, we need to pre-define the data element better used for our analysis to enhance the efficiency of data capture.

Each account's dataset will have to predict its exclusive labels or tags. They usually have more intense emotions in the perception of these labels. Besides, it is more efficient to crawl the specific datasets by digging these related friends with common hobbies rather than grab a user associated entire weibos, friends, or articles. The second layer datasets such as indirect relationship is also being considered as the mining scope. There are several valuable dimensions we may use in the following:

1. Social Network Effect.
2. Personal Interest/Tag Effect.
3. Brand Effect.
4. Domain know-how Effect.
5. Education/Job/Living Background Effect.
6. Religion Belief Effect.
7. Buying Behaviour/Consumption Concept Effect.
8. Blood Type/Constellation Effect.

5 Process of Profiling Analysis

The following will introduce the primary core analysis algorithms processed in several stages: Keywords and characteristics mapping, directly or indirectly, weight calculation, sketching portraits of target user, portrait visualization.

5.1 Keywords and Characteristic Mapping

Firstly we use kNN (K nearest neighbor algorithm) to filter the collected datasets. The key dimension we mention above at last chapter here need to be calculated as the query point. We gather k nearest neighbor datasets (k is a optimized constant.) extended from the query point and dismiss the others. As Fig. 2 shown, query point can be used as our key data dimension, then we group these datasets as the keyword clusters (Fig. 3).

After that, the keyword clusters would be executed in a matching process through a mapping library. If the keyword cluster does not match the corresponding characteristic, we still need to use fuzzy mapping to determine compliance with matching process. Another fuzzy-word repository will need to be built to help complete the comparison process.

5.2 Direct and Indirect Relation Weighs Calculation

Before the progress of the profiling scratch, we need to calculate the characteristic index by multiplying a coefficient. The coefficient differs from every character index and can be optimized by self-learning process. For the reason that the previous rounds

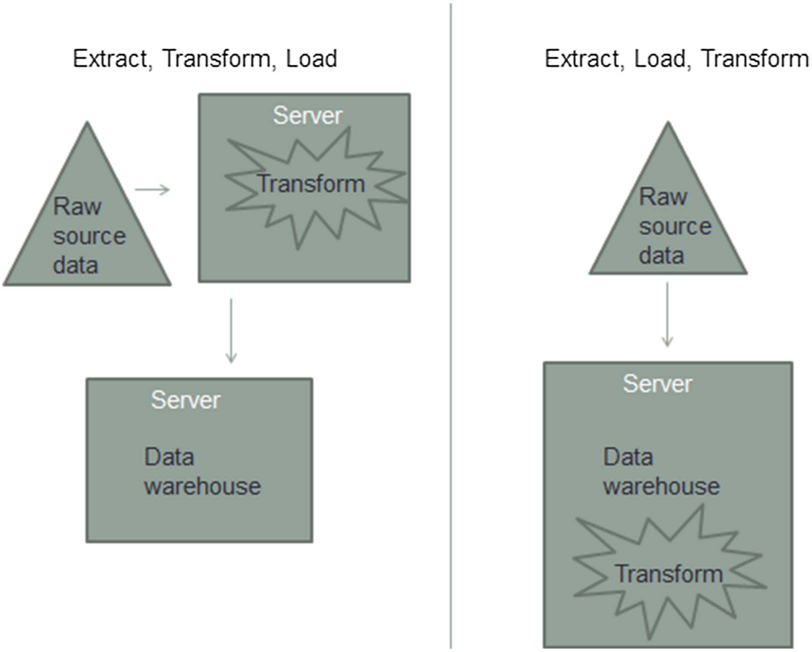


Fig. 2. Comparison between ETL and ELT process

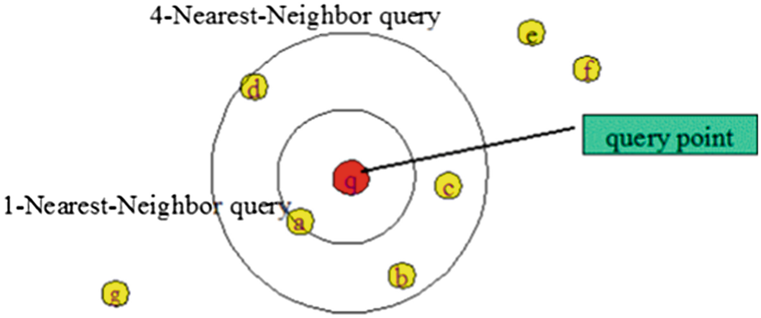


Fig. 3. Concept map of kNN algorithm

will not reflect the precise prediction, here we used Perceptron Learning Algorithm (PLA) to have self-optimized mechanism to improve the predictive coefficient. Therefore, these various indexes are ready to process the next stage.

5.3 Natural Language Processing

The Chinese sentences are inherently complex and difficult to be analyzed by general NLP. Firstly, we define a logical semantic in terms of English-like subject, verb, noun, and objective, etc. this logical semantic is used as an analyzer to structure a whole sentence. Next, we built up ontology to help us define the relation between the above entities. Besides, the purpose would be resolved in this stage. For example, there is a sentence ‘Could you please turn the radio off?’ In Chinese, the attitude might make the sentence as an imperative sentence instead of a question sentence. Another example is that ‘I don’t like John’s t-shirt today.’ The speaker may want to emphasize “John’s” which may imply he or she would admire another one’s t-shirt. Or if he or she was looking at that “t-shirt” he or she don’t like. In another possibility is that he or she just doesn’t like that John’s t-shirt and he or she usually like John’s dressing. In order to identify the emphasized word, we usually trace the previous context to position the key word or we would memorize the final result and correct the former error. The last one stage is the ontological processing level. In Chinese sentence, sometimes the “one” sentence is not enough to express the true meaning because this sort of sentence is aimed at elaborating the following sentence or at spreading out the main idea with the coming paragraph. Therefore, we need a hypothetical algorithm to derive the meaning of the text before and after. In general, we will define a key word library to help us organize and predict the meaning in one sentence, then predict the real meaning by grouped key words of each sentence. The popular words also need to be considered in the hypothetical algorithm. However, the popular word library still needs to be manually conducted and revised by monitoring the news or other media channels.

5.4 Profiling Scratch

There are different categories based on each personality traits, and we need to use filters in front of the analytical datasets in scratching out the personality profile. Since everyone has different brand preferences, hobbies, life value, etc. The character tags need to be refined and complied into a tag library. Each tag would have level 2 tags even deeper level to distinguish the extent of the impact on individual labels. Thus we use Random Forest Classification as the classification algorithm to classify and calculate multi-dimensional tags within the analytical datasets. By refining the deeper character indexes, the scratching profile is gradually portrayed [7], as the Fig. 4 shown, the classification algorithm will calculate each effect with the combination of all character index datasets.

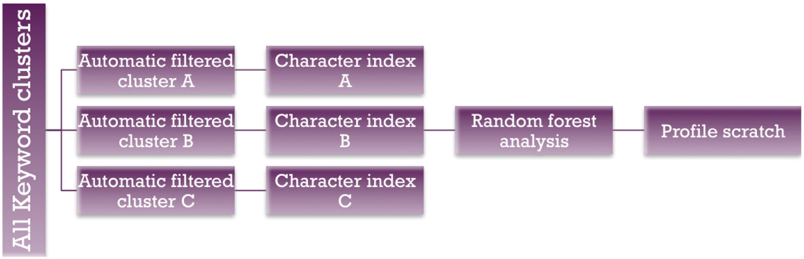


Fig. 4. Concept of engine calculation

5.5 Visualize the Target Profile

There are many ways to visualize the analyzed data. Since we think human beings must have various and complex characters within his/her personality. These characters may be shown explicitly or implicitly. Therefore, we use pie chart to demonstrate the probabilities of each character that may fairly perform the analyzed result (Fig. 5).

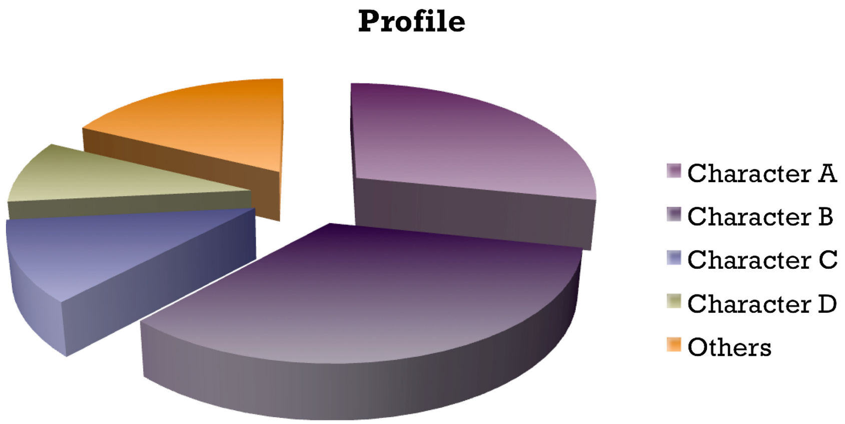


Fig. 5. Pie chart of target profile

6 Conclusion

Although human behavior is very complex and nearly unpredictable, we are still trying to build up the Profile Analyzer System to help us deeply understand the individual characters and scratching out the result in digits. In the coming research, we are focused on deeply analyze related groups the user joined and the related events the user involved. The extended information may not only enhance the accuracy of predictive analysis but also amend the previous analysis. We believe that the analyzed profiling results, demonstrated a portion of characters, can be applied in many business modules that may motivate the other business group to join our research findings or to help us deeply enhance our system. Finally, the proposed system has been applied to in real scenario with the dataset collected from Sina weibo and the experiment shows the analyzed result in a certain real case.

References

1. Weiss, S.H., Indurkhya, N.: Predictive Data Mining: A Practical Guide. Morgan Kaufmann Publishers, San Francisco (1998)
2. Piatetsky-Shapiro, G., Frawley, W.J.: Knowledge Discovery in Database. AAAI/MIT Press, Cambridge (1991)
3. Han, J.: Data Mining Concepts and Techniques. Morgan Kaufmann, San Diego (2006)

4. Miller, R.C., Bharat, K.: SPHINX: A framework for creating personal, site-specific web crawlers. In: Proceedings of the Seventh International World Wide Web Conference, pp. 119–130 (1998)
5. Henzinger, M., Heydon, A., Mitzenmacher, M., Najork, M.A.: measuring index quality using random walks on the web. In: Proceedings of the Eighth International World Wide Web Conference, pp. 213–225 (1999)
6. Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. In: Proceedings of OSDI (2004)
7. Duda, R.O., Hart, P.E., Stork, D.G.: Chapter 8. Pattern classification, 2nd edn. In: A Wiley-Interscience Publication (2001)
8. Astesiano, E., Zucca, E.: Parametric channels via label expressions in CCS. J. Theor. Comput. Sci. **33**, 45–64 (1984)

Analyzing Personality Traits Based on User Behavoirs of Using Microblog

Xin Fang(✉), Bing Li, and Wei-Hao Chang

SIGI Technologies Inc., Beijing, China
{singh.fang,Oliver.jang}@gmail.com,
libing_sigi@outlook.com

Abstract. This paper devotes to describe a novel way of making a good educated analysis about personality traits of users from Sina Weibo - the most popular microblog platform in Mainland China. Firstly, It will describe the current method of Social Media listening technology to analysis news, trends, stats & insights, which will also become the basement and macro factors of our new purposed framework indicated in this paper. Secondly, it will describe in details of our purposed solution to analysis each individual Weibo users behaviors and connections describe their personalities, hobbies, social conditions etc. Thirdly, the application and usage of this novel system will also be indicated in this paper, to describe the advantages and how it will help enterprise and individual users.

Keywords: Data mining · Micro-blog · Personality traits · Social marketing

1 Introduction

In recent years, the booming of social network has already changed the way people getting and sharing information. In China, microblog sites such as Sina Weibo, has now become one of the most important platforms and channels for a lot of people to obtain and transfer information. Tremendous amount of data is being generated everyday on Weibo platform, it includes User-Generated contents (posts, comments messages), user behaviors (following, repost, like), objective environment information (time, entrance, location), thus, monitoring and analysing these data and data changes could be used to identify social trends and user preference. In the mean time, enterprises also want to benefit from using social big data to have better understanding about target market or users to help make precision marketing strategies and it has brought a lot of new marketing methods, propaganda, and even changed the course of enterprise products [1].

At present, the usage of social big data analysis is mainly focus on mainstream social networking platforms by crawling the whole SNS data as the unit to implement public opinion monitoring, fuzzy trends prediction. There are also more and more companies especially large-scale companies, trying to use social big data analysis in order to improve the efficiency of Customer Relationship Management or helping to product marketing strategy. However, at present the analysis and prediction based on social platform is more limited in the English language on the social platforms like

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 773–779, 2014.
DOI: 10.1007/978-3-319-13186-3_69

Facebook and Twitter, and most of the time, the social media analysis platforms are mostly designed and developed specifically for large-scale enterprises and business groups, thus how to make the social big data mining in the service of more small and medium-sized enterprise and individual is a more practical subject.

At the same time, Microblog platforms are growing through an unprecedented boom in recent years. The number of Chinese Micro-blog registered users had over 1.3 billion by the end of 2013, there are millions of people contribute contents through various terminals to this platform and making it a major channel for disseminating information. Comparing with Facebook and Twitter, Microblog in China have a lot of differences, Chinese Weibo users prefer more to retweet posts and most of them treat Weibo platform as another media portal, so information would spread extremely fast on Weibo platform. So Weibo is a Time-saving and Cost-efficient way to help promote products and make online marketing for enterprises, companies like Coco-cola, Nestle, Durex have proved many classical cases of successful Weibo marketing.

This study focus on analysis individual characteristics of Microblog user, it will purpose a novel way to full range of analysing individual insights on Weibo, based on a self-evolving external environment by social media monitoring and trends prediction. In order to achieve that, we purposed a system called PAS- Profile Analyzer System, it generally composed by three parts: 1. A social media listening platform, which will continuously crawl the whole Weibo data, and deep analysis, all related trends and connections by specific queries of searching conditions (such as products, industries, company names etc.). 2. An individual social analytics tool to provide sentiment analysis or behavior analysis for each Weibo users, and it will contain features like: fake user identification, user influence analysing, general characterises prediction, social status speculate etc. 3. Both of two parts will finally take to a full report and a front-end display module to allow individual users and enterprise access to reuse. The system is aiming to give enterprise a better way to find real potential buyers and what they really want, and it will also help to get rid of annoying and inaccurate messages to individuals.

2 Related Work

Social networks are “part of social media which are applied to any kind of products and services and user generated content which includes conversation, articles, images or pictures, recipes, and anything that an individual share with others in their daily lives” [2]. Based on this, social network platforms are always major objects for research and enterprise to analysis people. There are lots of cases of using SNS to predict real life activities, Hua-Ping Zhang wrote a paper to deep research on how and what SNS will create deep influences, and come to a that “user content intention analysis has been performed to reveal users’ most concerns in their daily life” [3], Donghao Ren et al., even developed a visual analytics system to enable general Weibo users to analysis every Weibo events and geographical situational awareness, it could provide three models of dynamic graphing to show spread rout and start sourcing of one partially post. Reference [4] For business purposes, Cisco has already developed a Social CRM system for enterprise to quickly receive and response in real time every related phenomena and trends about their company or products, and be closer to their customers [5]. Other companies in China, like Admaster, Knowlesys and even the enterprise version of Sina Weibo added the similar features for their enterprise customers.

In addition of using microblog to do public opinion monitoring and trend prediction, there are also many researcher and products focusing on characterising individual users of SNS users. These implementations at present are mainly through the analysis of the behavior of user in the platform, in order to draw its activity and influence in the virtual society. In order to reduce the "Fake Users" in Micro-blogging system which may cause negative influences and restore the Micro-blogging a pureness environment, [6] Guanzhi Z. has purposed a mechanism of how to identify Fake followers based on their behaviors such as the frequency of making posts, Fans Numbers, Friends Numbers. Reference [7] Binlin C. has also purposed similar method of how to judge user influences on Chinese Microblog- Weibo, and filter fake followers. Reference [8] Additionally, Sina Weibo has recently release an application called micro-data to allow every Weibo user to analysis their own account to understand their influences on the platform including their Activity, the propagation force and coverage, and even provide them a chance to test their relationship with anyone on the platform including Big V (i.e. verified account of high-profile people on the internet) based on the "Six Degrees of Separation". Teresa Correa purposed a new model to pay attention on Text and behaviors generated by users to make a good educated guess on their traits based on the "BIG FIVE FRAME WORK" theory, it is more like a combination of "Human behavior, psychology, and data mining technology". Reference [10] While at the same time, IBM also release a research going even deeper, they analysis more than 10 million micro-blogging users, and announce that they could Undercover "Deep psychological profiles" of each individual users [9].

3 The Prediction Model of Micro-blogging User's Personalities

Based on theories and previous works introduced above, we would like to introduce a new system to achieve real-time public monitoring, trends prediction and individual insights, provide a full range of deep mining on every user's real need. In order to do this, the system will include following parts:

1. Social Listening Platform

The system could continually crawl data from Weibo platform by using data mining tools like WebHarvest, and it will also request data directly from Weibo open API.

Thus the system could analysis relevant trends, make predictions and early warning according to requests by enterprises based on the Weibo data generated during a long period of time to real time.

Through these vast amounts of data, the system will firstly retrieve the associated Weibo topics, characters, and events according to search queries, the system could monitor all relevant contents to provide a complete picture, and it may help enterprises to have better understanding about relevant topic or key messages spreading alongside. Then the system will automatically search related context to make deep analysis based on user operations like repost, comment, search keywords. After this step, the system will execute overlap filter information to find the relationship map and then import to a statistic analysis. Under this process, it could track the fastest rising topic, and predict possible news, or specifically track a product name to analysis where and how people discuses and response, and even it could analysis which of them has the most influences to others. These kind of prediction and reasoning will directly help enterprise to realize how their products create affections in public, and what can they improve to enlarge the impacts and find right person to promote. More importantly, this process will help to generate an environment parameter to help making future individual analysis. Consider the unique feature of Sina Weibo, Big Vs and some official account will have great influence on spreading events, we designed a modelling specifically to analysis the degree these Big V have may have influences, and it will help to forecast future event spreading rout and coverage.

3.1 An Individual Social Analytics Tool

The system also provide a tool to deep analysis individual Sina Weibo Users, based on their original profile, actions like following, repost and like, and also their friend circles. Through this process, it will come to a "user portrait" which is used to describe user preferences, hobbies, social status and other real life properties.

The development of the Internet has changed people's attitude to information privacy, Online social media users have begun to open part of their personal data to their friends or even public world, in order to know new friends or get better services, As the foundation of the social conduct, we call these data the user's Static Information.

This information on the Weibo platform could include: Registered user name/ID, location, birthday/sign, blood type, the registration time, gender, work/school users from losing data, and the widely use of tags by others, for example, favorite in food, movies, a otaku, etc. (Figs. 1 and 2).

Because of these information are normally input by user themselves, it can be treat as the basic data structure of user attributes, and carry on simple classification of them, but at the same time, the data also has deficiencies in high risk, false or inaccurate, so these data is only as a standard reference data, and later to obtain objective data with a weight comparison.

The most important step during this process is to make a longitudinal comparison and analysis the dynamic data generated during the process of long-term usage. Most of users traits are hidden in its routine operations and contents of their posts [9], so it will require a period of time of usage to execute text analysis and comparison of a longitudinal change. For example, we can infer from one user's 50 posts initiatively sent to Weibo about its most like punctuation marks, interactive friends, frequency of active in Weibo and even its location and using time can be deduced.

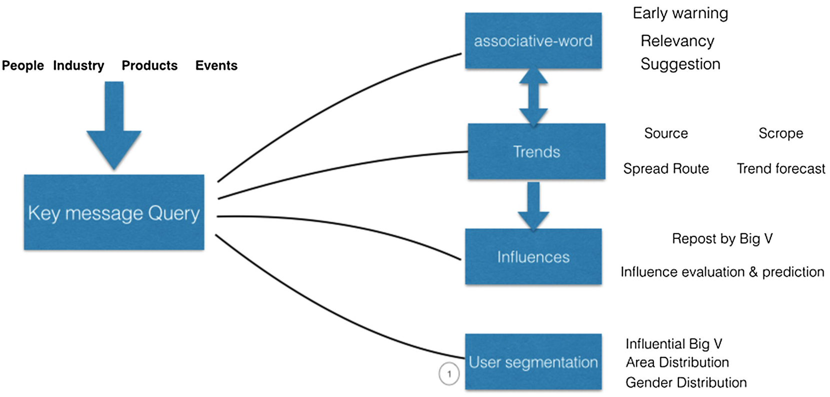


Fig. 1. Social listening platform

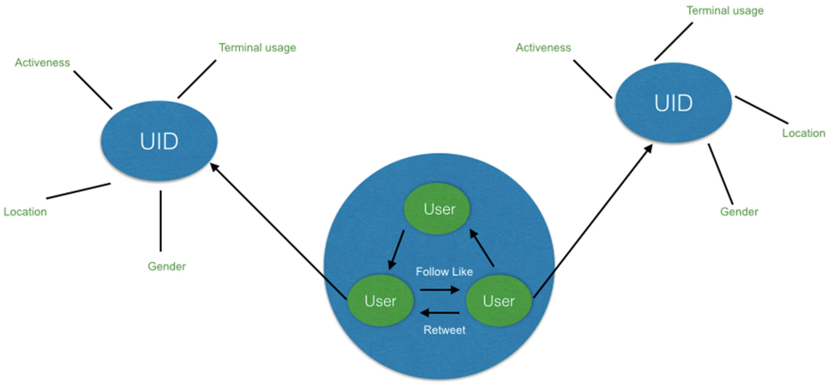


Fig. 2. Social analytics tool

4 Application of Analyzing Personality Traits of Weibo User

The system could deeply draw a big picture of trends and changes on particular area, based on mining every personal Weibo data, and Educated analysis individuals characteristics will be more conducive to help associated each user from the vast amounts of information in the Internet, thus enterprise could gain better understanding of their users, and push the right products and advertising in a more appropriate time; Every users on social networking can also easily find interested contents displayed in right places; Weibo itself can not only be a platform to communicate but also can become the center of origin to understand people.

After the deep analysis of every Weibo users' preferences and traits enterprises could quickly find their target users and increase the accuracy of their Ad. For example, for a terminal company, to learn how to quickly locate its products to suitable crowd and put in the right model to each group of people to grasp their interest is an important marketing measure, one of current solution now is to promote several similar Ads, and publish to the public, which is very inflexible, and unpredictable, another way is based on the simple user attention and buying behavior to execute simple similarity matching promotion; Based on the previous analysis on Weibo group and individual, the PAS system can effectively help enterprises to specify more intelligent advertising promotion strategy; Big data analysis based on groups of people may help enterprises to quickly find their target users preferences to know what information and unique features they may concerning about, and it also help to lock their competitors' products about their future influences on Weibo, also enterprise would know the most influential people to their target users, and maximize the utilization of resources. For each user analysis, it can include the status of work/study, location, and even analysis their terminals of sending Weibo messages to indicate the change of their mobile terminal, with general concern of user's social status, revenue forecasts, so when they have changed a new phone, the system can help enterprises to know after how long the user will most likely to replace the phone and which model they may interested, then push more accurate advertisement and directional information, thus to greatly increase the purchase willing of their potential users.

At the normal user's aspect, with the development of the Internet, a large amount of data and content have already made all users into information slaves, how to help users to intelligent filter information based on their behavior and preferences will be a very important direction of future development on the mobile Internet. Based on the characteristics of users micro-blogging data after scanning, it can effectively learn what contents may attract each user's focus, And by analysing their relationship chain, it could understand their social status, their real-life range, and even predict where it might go and purchase products. More additional, through the use of the terminal equipment of the positioning system to understand the user's position, time, etc., on the basis of these data it can almost understanding each user's needs of contents and apps, and deliver them in the right place.

5 Conclusion

This paper purposed a new method of analysing micro-blogging data to help enterprises make better marketing methods by having better understanding about their target industry, market, users and even competitors. It described a system called PAS to firstly collect mass of data from Sina Weibo, the most popular micro-blogging site, and to generate the environment to predict the trends and public activities, and help to find the big picture for companies to understand their brand, products, or marketing results. After that, PAS also could scan and analysis every individual users to find the real valuable potential customers and people who may have influence to them which will dramatically improve the accuracy and effectives for companies to promote products or hold activities. Finally in this report, we indicate some usage aspects of our system, which may not only help companies to do CRM or product promotion but also help every normal users to understand their primary images on the internet, and contribute to other content providers of optimising information to every normal internet users.

References

1. Schmidt, R.: Social data for product innovation, marketing and customer relations. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBIP, vol. 132, pp. 234–245. Springer, Heidelberg (2013)
2. Siricharoen, W.: Social media, how does it work for business? Int. J. Innov. Manag. Technol. **3**(4), 476–479 (2012)
3. Hua-Ping, Z., Rui-Qi, Z., Yan-Ping, Z., Bao-Jun, M.: Big data modeling and analysis of microblog ecosystem. Int. J. Autom. Comput. **11**, 119–127 (2013)
4. Ren, D., Zhang, X., Wang, Z, Li, J., Yuan, X.: WeiboEvents: a crowd sourcing weibo visual analytic system. In: IEEE Pacific Visualization Symposium (Pacific Vis 2014), Visualization Notes, Yokohama, Japan, (2014)
5. Cisco.com, A report on Cisco Social Minor (2011). http://www.cisco.com/en/US/products/ps11349/index.html. Accessed Jan 2011
6. Mondal, M., Viswanath, B., Clement, A., et al.: Limiting large-scale crawls of social networking sites. ACM SIGCOMM Comput. Commun. Rev. **41**(4), 398–399 (2011). ACM
7. Zhang, G., Bie, R.: Discovering high-quality users from sina weibo based on trust transfer model. J. Comput. Inf. Syst. **9**(16), 6467–6478 (2013)
8. Binlin, C., Jianming, F., Jingwei, H.: Detecting zombie followers in sina microblog based on the number of common friends. Int. J. Adv. Comput. Technol. **5**(2), 612–620 (2013)
9. Correa, T., Bachmann, I., Hinsley, A.W., et al.: Personality and Social Media Use (2013)
10. Corley, C.D., Cook, D.J., Mikler, A.R., et al.: Text and structural data mining of influenza mentions in web and social media. Int. J. Environ. Res. Public Health **7**(2), 596–615 (2010)

BigSAM: Mining Interesting Patterns from Probabilistic Databases of Uncertain Big Data

Fan Jiang, Carson Kai-Sang Leung(✉), and Richard Kyle MacKinnon

University of Manitoba, Winnipeg, MB, Canada
kleung@cs.umanitoba.ca

Abstract. Nowadays, high volumes of valuable uncertain data can be easily collected or generated at high velocity in many real-life applications. Mining these uncertain Big data is computationally intensive due to the presence of existential probability values associated with items in every transaction in the uncertain data. Each existential probability value expresses the likelihood of that item to be present in a particular transaction in the Big data. In some situations, users may be interested in mining all frequent patterns from these uncertain Big data; in other situations, users may be interested in only a tiny portion of these mined patterns. To reduce the computation and to focus the mining for the latter situations, we propose a tree-based algorithm that (i) allows users to express the patterns to be mined according to their intention via the use of constraints and (ii) uses MapReduce to mine uncertain Big data for only those frequent patterns that satisfy user-specified constraints. Experimental results show the effectiveness of our algorithm in mining probabilistic databases of uncertain Big data.

1 Introduction and Related Works

Nowadays, high volumes of valuable data are easily collected or generated in various real-life application areas such as bioinformatics, e-commerce, healthcare, mobile sensing, security, and social networks. This leads us into the new era of Big data [20]. Intuitively, *Big data* refer to high-velocity, high-value, and/or high-variety data with volumes beyond the ability of commonly-used software to capture, curate, manage, and process within a tolerable elapsed time. Hence, new forms of processing data are needed to deliver high veracity (and low vulnerability) and to enable enhanced decision making, insight, knowledge discovery, and process optimization. This drives and motivates research and practices in business analytics and optimization, which require techniques like Big data mining and analytics [12,14]. Having developed systematic or quantitative processes to mine and analyze Big data allows us to continuously or iteratively explore, investigate, and understand the past business performance so as to gain new insight and drive business planning.

To handle Big data, researchers proposed the use of a high-level programming model—called *MapReduce*—to process high volumes of data by using parallel and

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 780–792, 2014.
DOI: 10.1007/978-3-319-13186-3_70

distributed computing [25] on large clusters or grids of nodes (i.e., commodity machines), which consist of a master node and multiple worker nodes. As implied by its name, MapReduce involves two key functions: "map" and "reduce". An advantage of using the MapReduce model is that users only need to focus on (and specify) these map and reduce functions—without worrying about implementation details for (i) partitioning the input data, (ii) scheduling and executing the program across multiple machines, (iii) handling machine failures, or (iv) managing inter-machine communication. Over the past few years, several algorithms have been proposed to use the MapReduce model—which applies distributed or parallel computing—for different Big data mining and analytics tasks [5,10]. Examples of these tasks include clustering [6], outlier detection [9], and structure mining [24]. An equivalently important mining and analytics task is *pattern mining* [3,18,23], which aims to analyze valuable data for the discovery of implicit, previously unknown, and potentially useful knowledge in the form of patterns. For example, *frequent patterns* help reveal collections of popular merchandise items, frequently co-occurring objects, or frequently co-located events.

Since the introduction of pattern mining [1], numerous algorithms have been proposed to mine *precise* data such as shopper market basket transaction databases. With these databases, users definitely know whether an item is present in—or is absent from—a transaction. In this notion, each item in a transaction t_j in databases of precise data can be viewed as an item with a 100 % likelihood of being present in t_j. However, data in many real-life applications are riddled with uncertainty [2,16,19]. It is partially due to inherent measurement inaccuracies, sampling and duration errors, network latencies, and intentional blurring of data to preserve anonymity. Hence, users are usually uncertain about the presence or absence of items. As a concrete example, a meteorologist may suspect (but cannot guarantee) that severe weather phenomena will develop during a thunderstorm. The uncertainty of such suspicions can be expressed in terms of existential probability. For instance, a thunderstorm may have a 60 % likelihood of generating hail, and only a 15 % likelihood of generating a tornado, regardless of whether or not there is hail. To handle uncertain data, several pattern mining algorithms—such as the U-Apriori [4], UF-growth [15], and PUF-growth [16] algorithms—were proposed in previous PAKDD conferences.

Furthermore, in many real-life applications, users may have some particular phenomena in mind on which to focus the mining (e.g., a meteorologist may want to find only those weather records about thunderstorms with hail). However, the aforementioned algorithms mine patterns without user focus. Consequently, users often need to wait for a long period of time for numerous patterns, out of which only a tiny fraction may be interesting to the users. Hence, *constrained pattern mining* [11]—which aims to find *valid* patterns (i.e., patterns that satisfy the user-defined constraints)—is needed.

A natural question to ask is: Can we use MapReduce to perform constrained pattern mining from uncertain Big data? In response to this question, we propose an algorithm—called BigSAM—to mine uncertain Big data for frequent patterns that satisfy a particular type of user-specified constraints called succinct anti-monotone (SAM) constraints. Our algorithm discovers patterns from probabilistic databases of uncertain Big data in a pattern-growth fashion for Big data analytics. Such an algorithm—which is a non-trivial integration of (i) mining for patterns, (ii) mining from uncertain data, (iii) mining from Big data, and (iv) mining with constraints—is our key contribution of this paper.

The remainder of this paper is organized as follows. The next section gives background about uncertain data, SAM constraints, and the MapReduce model. In Sect. 3, we propose our BigSAM algorithm for mining uncertain Big data for patterns that satisfy the user-specified SAM constraints using MapReduce. Evaluation results and conclusions are presented in Sects. 4 and 5, respectively.

2 Background

In this section, we give some background information about (i) uncertain data, (ii) SAM constraints, and (iii) the MapReduce model.

2.1 Uncertain Data: Existential Probability and Expected Support

Let (i) `Item` be a set of m domain items and (ii) $X = \{x_1, x_2, \ldots, x_k\}$ be a pattern comprising k items (i.e., a *k-itemset*), where $X \subseteq$ `Item` and $1 \leq k \leq m$. Then, each item x_i in a transaction $t_j = \{x_1, x_2, \ldots, x_h\} \subseteq$ `Item` in a probabilistic database of uncertain data is associated with an *existential probability value* $P(x_i, t_j)$ [13], which represents the likelihood of the presence of x_i in t_j. Note that $0 < P(x_i, t_j) \leq 1$. The existential probability $P(X, t_j)$ of a pattern X in t_j is the product of the corresponding existential probability values of every item x within X (when these items are independent) [13]: $P(X, t_j) = \prod_{x \in X} P(x, t_j)$.

The *expected support* $expSup(X)$ of X in the probabilistic database is the sum of $P(X, t_j)$ over all n transactions in the database:

$$expSup(X) = \sum_{j=1}^{n} P(X, t_j) = \sum_{j=1}^{n} \left(\prod_{x \in X} P(x, t_j) \right), \tag{1}$$

where $P(x, t_j)$ is the existential probability value of item x in transaction t_j. With this definition of expected support, existing tree-based algorithms [8] such as UF-growth [15] and PUF-growth [16] mine frequent patterns from a probabilistic database of uncertain data as follows. The algorithms first scan the database once to compute the expected support of all domain items (i.e., singleton itemsets). Infrequent items are pruned as their extensions/supersets are guaranteed to be infrequent. The algorithms then scan the database a second time to insert all transactions (with only frequent items) into a tree (i.e., UF-tree [15] or PUF-tree [16]). Each node in the tree captures (i) an item x, (ii) its existential probability value $P(x, t_j)$, and (iii) its occurrence count. At each step during the mining process, the frequent patterns are expanded recursively.

Given such a database and a user-specified minimum support threshold *minsup*, the research problem of *frequent pattern mining from uncertain data* is to discover from the probabilistic database of uncertain data all those *frequent* patterns (i.e., patterns having expected support $\geq$ *minsup*).

2.2 Succinct Anti-monotone (SAM) Constraints

To mine interesting patterns from precise data, the Constrained Apriori (CAP) framework [21] allows the user to specify his interest via the use of SQL-style constraints to guide the mining process so that only those frequently occurring sets of shopper basket items satisfying the user constraints are returned. This avoids unnecessary computation for mining those uninteresting frequent patterns. Examples of user-specified constraints include (i) $max(\text{X}.Price) \leq \$500$, which expresses the user's interest in finding every frequent pattern X such that the maximum price of all shopper market basket items in X is at most $500, and (ii) $X.Type \neq snack$, which expresses the user's interest in finding every frequent pattern X such that every shopper market basket item in X is not a snack item.

In general, user-specified constraints can be categorized into several overlapping classes according to the properties that they possess. The two aforementioned constraints in particular can be categorized into a popular class of constraints called *succinct anti-monotone* (*SAM*) *constraints*, which possess the properties of both *succinctness* and *anti-monotonicity*.

Definition 1 (Succinctness [11]). Let `Item` be the set of m domain items. Then, an itemset $SS_j \subseteq \texttt{Item}$ is a *succinct set* if SS_j can be expressed as a result of selection operation σ_p (`Item`), where σ is the usual SQL-style selection operator and p is a selection predicate. A powerset of items $SP \subseteq 2^{\texttt{Item}}$ is a *succinct powerset* if there is a fixed number of succinct sets $SS_1, \ldots, SS_k \subseteq \texttt{Item}$ such that SP can be expressed in terms of the powersets of $SS_1, \ldots, SS_k$ using set union and/or set difference operators. A constraint C is ***succinct*** provided that the set of patterns satisfying C is a succinct powerset. □

Definition 2 (Anti-monotonicity [11]). A constraint C is ***anti-monotone*** if and only if all subsets of a pattern satisfying C also satisfy C. □

2.3 The MapReduce Programming Model

MapReduce [7] is a high-level programming model for processing vast amounts of data. Usually, MapReduce uses parallel and distributed computing on clusters or grids of nodes (i.e., computers). The ideas behind MapReduce can be described as follows. As implied by its name, MapReduce involves two key functions: "map" and "reduce". The input data are read, divided into several partitions (subproblems), and assigned to different processors. Each processor executes the *map function* on each partition (subproblem). The map function takes a pair of $\langle key, value \rangle$ data and returns a list of $\langle key, value \rangle$ pairs as an intermediate result:

$$\text{map: } \langle key_1, value_1 \rangle \mapsto \text{list of } \langle key_2, value_2 \rangle,$$

where (i) key_1 & key_2 are keys in the same or different domains, and (ii) $value_1$ & $value_2$ are the corresponding values in some domains. Afterwards, these pairs are shuffled and sorted. Each processor then executes the *reduce function* on (i) a single key value from this intermediate result together with (ii) the list of all values that appear with this key in the intermediate result. The reduce function "reduces"—by combining, aggregating, summarizing, filtering, or transforming—the list of values associated with a given key (for all k keys) and returns a list of k values:

$$\text{reduce: } \langle key_2, \text{list of } value_2 \rangle \mapsto \text{list of } value_3,$$

or returns a single (aggregated or summarized) value:

$$\text{reduce: } \langle key_2, \text{list of } value_2 \rangle \mapsto value_3,$$

where (i) key_2 is a key in some domains, and (ii) $value_2$ & $value_3$ are the corresponding values in some domains. Examples of MapReduce applications include the construction of an inverted index as well as the word counting of a document.

Early works on MapReduce focused either on data processing [7] or on some data mining tasks other than frequent pattern mining (e.g., outlier detection [9], structure mining [24]). Recently, three Apriori-based algorithms called SPC, FPC and DPC [17] were proposed to mine frequent patterns from *precise data*. Like these three algorithms, our proposed BigSAM algorithm also uses MapReduce. However, unlike these three algorithms (which mine frequent patterns from *precise data* using the *Apriori-based approach*), our proposed BigSAM mines frequent patterns from *uncertain data* using a *tree-based approach*. The search/solution space for pattern mining from uncertain data is much larger than that for pattern mining from precise data due to the presence of the existential probabilities.

Moreover, a parallel randomized algorithm called PARMA [22] was proposed to mine *approximations* to the top-k frequent patterns and association rules from *precise data* using MapReduce. Although PARMA and our BigSAM algorithm both use MapReduce, one key difference between the two algorithms is that we aim to mine for *truly frequent* (instead of approximately frequent) patterns. Another key difference is that we mine from *uncertain data* (instead of precise data). The third key difference is that we mine with constraints so that we focus our computation on finding only those frequent patterns *that satisfy the use-specified SAM constraints* (instead of all frequent patterns).

3 Our BigSAM Algorithm that Mines Uncertain Big Data for Frequent Patterns Satisfying SAM Constraints

Given (i) a probabilistic database of uncertain Big data, (ii) a user-specified SAM constraint, and (iii) a user-specified minimum support threshold *minsup*, our proposed BigSAM algorithm uses the MapReduce programming model to mine uncertain Big data—in a tree-based pattern-growth fashion—for all patterns satisfying the SAM constraint and having expected support ≥ *minsup* (i.e., *valid frequent patterns*).

Recall from Sect. 2.2 that users can express their interests by specifying constraints. Our BigSAM algorithm does not confine the user-specified constraints to only shopper market basket items. We allow users to specify constraints that can be imposed on items, events, or objects in other domains. For example, constraint $C_1 \equiv max(X.Temperature) \leq 20°\text{C}$ expresses the meteorologist's interest in finding every frequent pattern X such that the maximum temperature of all meteorological records in a pattern X is at most 20°C. In other domain (e.g., healthcare sector), constraint $C_2 \equiv min(X.WBC) \geq 4.3 \times 10^9/\text{L}$ expresses the physician's interest in finding every group X such that the minimum white blood cell (WBC) counts among all patients in X is at least $4.3 \times 10^9/\text{L}$. For mobile sensing, constraint $C_3 \equiv X.Location{=}Tainan$ expresses the user interest in finding every frequent pattern X such that all events in X are held in the city of Tainan; constraint $C_4 \equiv X.Location{=}(Tainan \vee Kaohsiung)$ expresses the user interest in finding every frequent pattern X such that all events in X are held in Tainan or Kaohsiung. Constraint $C_5 \equiv X.Location{\neq} Winnipeg$ expresses the user interest in finding every frequent pattern X such that all events in X are held outside Winnipeg.

We observed that, due to anti-monotonicity, if a pattern does not satisfy the SAM constraints, all its supersets are guaranteed not to satisfy the SAM constraints. Thus, any pattern that does not satisfy the SAM constraints can be pruned. Moreover, due to succinctness, we can precisely enumerate all and only those patterns that satisfy the constraints by using a member generating function. For example, the set of patterns satisfying $C_1 \equiv max(X.Temperature) \leq 20°\text{C}$ is a succinct powerset. Thus, the set of patterns satisfying C_1 can be expressed as $2^{\sigma_{Temperature \leq 20°\text{C}}(\texttt{Item})}$. The corresponding member generating function can be represented as $\{X \mid X \subseteq \sigma_{Temperature \leq 20°\text{C}}(\texttt{Item})\}$, which precisely enumerates all and only those *valid* patterns (i.e., patterns that satisfy C_1): All these patterns must comprised only items with individual temperature $\leq 20°\text{C}$. Consequently, valid frequent patterns for C_1 would be those frequent ones among the valid patterns satisfying C_1.

3.1 Exploiting SAM Constraints in the Reduce Function

With the above observations, our BigSAM algorithm exploits the properties of succinctness and anti-monotonicity. The key idea of our algorithm—which uses two sets of the map and reduce functions to mine uncertain Big data for frequent patterns satisfying SAM constraints—can be described as follows. BigSAM first reads high volumes of uncertain Big data. As each item in the volumes is associated with an existential probability, BigSAM aims to compute the expected support of all domain items (i.e., singleton patterns) by using MapReduce. The expected support of any pattern can be computed by using Eq. (1). Moreover, for any singleton pattern $\{x\}$, such an equation can be simplified to become the following:

$$expSup(\{x\}) = \sum_{j=1}^{n} P(x, t_j), \tag{2}$$

where $P(x, t_j)$ is the existential probability of item x in transaction t_j. Specifically, BigSAM divides the uncertain data into several partitions and assigns them to different processors. The *map function* receives ⟨transaction ID, content of that transaction⟩ as input. For every transaction t_j, the map function emits an $\langle x, P(x, t_j)\rangle$ pair for each item $x \in t_j$ (cf. when mining *precise* data, each occurrence leads to an actual support of 1 and would yield a corresponding $\langle x, 1\rangle$ pair). When mining *uncertain* data, the occurrence of x can be different from the expected support of x. For instance, consider an item a with existential probability value of 0.9 that appears only in transaction t_1. Its expected support may be higher than item b that appears seven times but with an existential probability value of 0.1 in each appearance. Then, $expSup(\{a\}) = 0.9 > 0.7 = expSup(\{b\})$. Hence, instead of emitting $\langle x, 1\rangle$ for each occurrence of $x \in t_j$, BigSAM emits $\langle x, P(x, t_j)\rangle$ for each occurrence of $x \in t_j$. In other words, the map function can be specified as follows:

```
For each transaction $t_j \in$ partition of the uncertain Big data do
    for each item $x \in t_j$ do
        emit $\langle x, P(x, t_j)\rangle$.
```

This results in a list of different $\langle x, P(x, t_j)\rangle$ pairs.

Afterwards, these $\langle x, P(x, t_j)\rangle$ pairs are shuffled and sorted. Each processor then executes the *reduce function* on the shuffled and sorted pairs to apply constraint checking on every item x and obtain the expected support only for valid x (i.e., $\{x\}$ that satisfies C_{SAM}). In other words, the reduce function can be specified as follows:

```
For each $x \in \{\langle x$, list of $P(x, t_j)\rangle\}$ do
    if $\{x\}$ satisfying $C_{\text{SAM}}$ then
        set $expSup(\{x\}) = 0$;
        for each $P(x, t_j) \in$ list of $P(x, t_j)$ do
            $expSup(\{x\}) = expSup(\{x\}) + P(x, t_j)$.
        if $expSup(\{x\}) \geq minsup$ then emit $\langle \{x\}, expSup(\{x\})\rangle$.
```

To recap, when using a high-level description, this first set of "map" and "reduce" functions can be defined as follows:

$$\begin{aligned} \text{map}_1\colon & \ \langle \text{ID of transaction}\, t_j, \text{content of } t_j\rangle \mapsto \text{list of } \langle x, P(x, t_j)\rangle, \\ \text{reduce}_1\colon & \ \langle x, \text{list of } P(x, t_j)\rangle \mapsto \langle \text{valid frequent}\{x\}, expSup(\{x\})\rangle. \end{aligned}$$

Here, the master node first reads and divides uncertain Big data in partitions. The worker node corresponding to each partition then outputs $\langle x, P(x, t_j)\rangle$ pairs for each domain item x. The reduce function then sums all existential probability values of x for each x to compute its expected support in the probabilistic database of uncertain Big data.

Example 1. Consider a probabilistic database of uncertain data with auxiliary information as shown in Table 1. Let (i) $C_1 \equiv max(X.Temperature) \leq 20\,^{\circ}\text{C}$, which means that domain items a, b, c, d (but not e) satisfy C_{SAM}, and (ii) *minsup*=1.0 Then, from transaction t_1, the map function outputs $\langle a, 0.7\rangle, \langle b, 1.0\rangle$, $\langle c, 0.8\rangle$ and $\langle e, 0.9\rangle$. Similarly, from transaction t_2, the map function outputs

Table 1. A sample set of uncertain Big data (with auxiliary information)

TID	Content
t_1	$\{a$:0.7, b:1.0, c:0.8, e:0.9$\}$
t_2	$\{a$:0.9, b:1.0, c:0.6, e:0.7$\}$
t_3	$\{a$:0.8, c:0.2, d:0.8$\}$

Item	Temp.	Item	Temp.
a	5°C	d	20°C
b	10°C	e	25°C
c	15°C		

$\langle a, 0.9\rangle$, $\langle b, 1.0\rangle$, $\langle c, 0.6\rangle$ and $\langle e, 0.7\rangle$; from transaction t_3, the map function outputs $\langle a, 0.8\rangle, \langle c, 0.2\rangle$ and $\langle d, 0.8\rangle$. These pairs are then shuffled and sorted. The reduce function reads $\langle a, [0.7, 0.9, 0.8]\rangle$, $\langle b, [1.0, 1.0]\rangle$, $\langle c, [0.8, 0.6, 0.2]\rangle$, $\langle d, [0.8]\rangle$ & $\langle e, [0.9, 0.7]\rangle$, and outputs $\langle a, 2.4\rangle, \langle b, 2.0\rangle$ &$\langle c, 1.6\rangle$ (i.e., valid frequent items and their corresponding expected support). Note that the reduce function does not sum the existential probabilities values for item e, let alone compute its expected support, because $\{e\}$ does not satisfy C_{SAM}. Although the reduce function sums the existential probabilities values for item d (as it satisfies C_{SAM}), it does not output its expected support because d is infrequent. □

3.2 Exploiting SAM Constraints in the Map Function

Alternatively, to handle the user-specified SAM constraint C_{SAM}, we can push C_{SAM} into the *map function* so that we only emit $\langle x, P(x, t_j)\rangle$ for each item $x \in t_j$ that satisfies C_{SAM}. See the corresponding map function:

```
For each transaction $t_j \in$ partition of the uncertain Big data do
    for each item $x \in t_j$ and $\{x\}$ satisfies $C_{\text{SAM}}$ do
        emit $\langle x, P(x, t_j)\rangle$.
```

This results in a list of different $\langle$valid $x, P(x, t_j)\rangle$ pairs.

Afterwards, these $\langle$valid $x, P(x, t_j)\rangle$ pairs are shuffled and sorted. Each processor then executes the *reduce function* on the shuffled and sorted pairs to obtain the expected support of x:

```
For each $x \in \{\langle$valid $x$, list of $P(x, t_j)\rangle\}$ do
    set $expSup(\{x\}) = 0$;
    for each $P(x, t_j) \in$ list of $P(x, t_j)$ do
        $expSup(\{x\}) = expSup(\{x\}) + P(x, t_j)$.
    if $expSup(\{x\}) \geq minsup$ then emit $\langle\{x\}, expSup(\{x\})\rangle$.
```

To recap, when using a high-level description, this alternative first set of "map" and "reduce" functions can be defined as follows:

$$\begin{aligned} \text{map}_{1'}\text{:} \quad & \langle \text{ID of } t_j, \text{content of } t_j\rangle \mapsto \text{list of } \langle \text{valid } x, P(x, t_j)\rangle, \\ \text{reduce}_{1'}\text{:} \quad & \langle \text{valid } x, \text{list of } P(x, t_j)\rangle \mapsto \langle \text{valid frequent } \{x\}, expSup(\{x\})\rangle. \end{aligned}$$

Example 2. Revisit Example 1. Again, let (i) domain items a, b, c, d (but not e) satisfy C_{SAM} and (ii) *minsup*=1.0. Then, from transaction t_1, the map function outputs $\langle a, 0.7\rangle$, $\langle b, 1.0\rangle$ and $\langle c, 0.8\rangle$. Similarly, from transaction t_2, the map function outputs $\langle a, 0.9\rangle$, $\langle b, 1.0\rangle$ and $\langle c, 0.6\rangle$; from transaction t_3, the map function outputs $\langle a, 0.8\rangle, \langle c, 0.2\rangle$ and $\langle d, 0.8\rangle$. Note that constraint checking was brought from the reduce function (as in Example 1) to the map function, which does not output the existential probability values for item e because $\{e\}$ does not satisfy C_{SAM}. All the output pairs are then shuffled and sorted. Afterwards, the reduce function reads $\langle a, [0.7, 0.9, 0.8]\rangle$, $\langle b, [1.0, 1.0]\rangle$, $\langle c, [0.8, 0.6, 0.2]\rangle$ & $\langle d, [0.8]\rangle$, and outputs $\langle a, 2.4\rangle, \langle b, 2.0\rangle$ & $\langle c, 1.6\rangle$ (i.e., valid frequent items and their corresponding expected support). Note that, although the reduce function sums the existential probabilities values for item d (as it satisfies C_{SAM}), it does not output its expected support because d is infrequent. □

As observed from the above two examples, exploiting SAM constraints in the reduce function requires fewer constraint checks because it only checks at most m domain items to see if they satisfy C_{SAM}. In contrast, exploiting SAM constraints in the map function checks all occurrences of items in every transaction in the Big data set, which are normally $\gg m$. Hence, the former is time-efficient when the data set consisting of only a few domain items such as DNA or RNA sequences in bioinformatics. On the other hand, the latter requires less bookkeeping because it emits $\langle \text{valid } x, P(x, t_j)\rangle$ only for those items that satisfy C_{SAM}. Hence, it is space-efficient when high volumes of data come at a high velocity such as data streams.

3.3 Mining Valid Frequent Non-singleton Patterns

Once our BigSAM algorithm finds valid frequent singleton patterns (with their associated expected support), it rereads each transaction in the probabilistic database of uncertain Big data to form an $\{x\}$-projected database (i.e., a collection of transactions containing x) for each valid frequent item x in the list returned by the first reduce function. Due to the succinctness &anti-monotonicity, all valid patterns must comprise only valid singleton items. Hence, our BigSAM algorithm keeps only those valid singleton items in each $\{x\}$-projected database. The worker node corresponding to each projected database then (i) builds appropriate local trees (e.g., UF-trees or PUF-trees)—based on the projected database assigned to the node—to mine every valid frequent pattern X of higher cardinality k (i.e., k-itemsets for $k \geq 2$) and (ii) outputs $\langle X, \mathit{expSup}(X)\rangle$ (i.e., every valid frequent pattern X with its expected support). In other words, BigSAM executes the second set of "map" and "reduce" functions as follows:

$$\begin{aligned}\text{map}_2\text{: } &\langle \text{ID of } t_j, \text{content of } t_j\rangle\\ &\mapsto \text{list of } \langle \text{valid frequent}\{x\}, \text{prefix of } t_j \text{that ends with } x\rangle,\\ \text{reduce}_2\text{: } &\langle \text{valid frequent}\{x\}, \{x\} - \text{projected database}\rangle\\ &\mapsto \text{list of } \langle \text{valid frequent pattern } X, \mathit{expSup}(X)\rangle.\end{aligned}$$

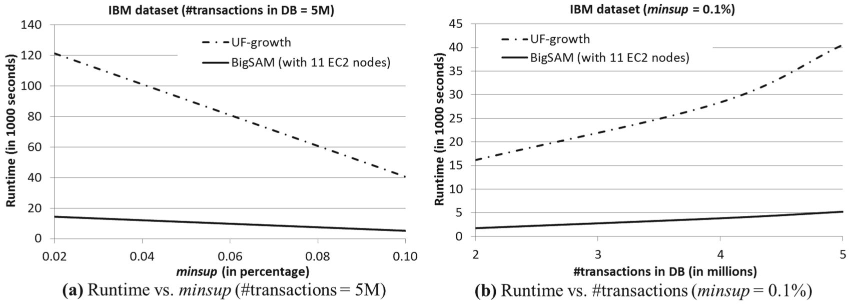
Example 3. Continue with Example 1 or 2. After reading t_1, BigSAM emits $\langle b, \{a{:}0.7, b{:}1.0\}\rangle$ & $\langle c, \{a{:}0.7, b{:}1.0, c{:}0.8\}\rangle$. BigSAM also emits $\langle b, \{a{:}0.9, b{:}1.0\}\rangle$ & $\langle c, \{a{:}0.9, b{:}1.0, c{:}0.6\}\rangle$ after reading t_2, and emits $\langle c, \{a{:}0.8, c{:}0.2\}\rangle$ after reading t_3. Note that the map function of BigSAM does not emit any pairs containing infrequent item d or invalid item e. After all the emitted pairs are shuffled and sorted, the reduce function of BigSAM then forms the $\{b\}$-projected database (comprising $\{a$:0.7, b:1.0$\}$ & $\{a$:0.9, b:1.0$\}$) and the $\{c\}$-projected database (comprising $\{a$:0.7, b:1.0, c:0.8$\}$, $\{a$:0.9, b:1.0, c:0.6$\}$ & $\{a$:0.8, c:0.2$\}$), from which valid frequent patterns $\{a, b\}$:1.8, $\{a, c\}$:1.26, $\{a, b, c\}$:1.1 and $\{b, c\}$:1.4 are found. □

4 Experimental Results

To evaluate our proposed BigSAM algorithm in mining uncertain Big data for frequent patterns that satisfy user-specified SAM constraints, we used different datasets—which include real-life datasets (e.g., accidents, connect4, and mushroom) from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/) and the FIMI repository (http://fimi.ua.ac.be/). We also used IBM synthetic data, which were generated using the IBM Quest Dataset Generator [1]. The generated data ranges from 2M to 5M transactions with an average transaction length of 10 items from a domain of 1K items. As these datasets originally contained only precise data, we assigned to each item in every transaction an existential probability value in the range (0,1]. All experiments were run on either (i) a single machine with an Intel Core i7 4-core processor (1.73 GHz) and 8 GB of main memory running a 64-bit Windows 7 operating system, or (ii) the Amazon EC2 cluster—specifically, 11 `m2.xlarge` computing nodes (http://aws.amazon.com/ec2/). All versions of the algorithm were implemented in the Java programming language. The stock version of Apache Hadoop 0.20.0 was used.

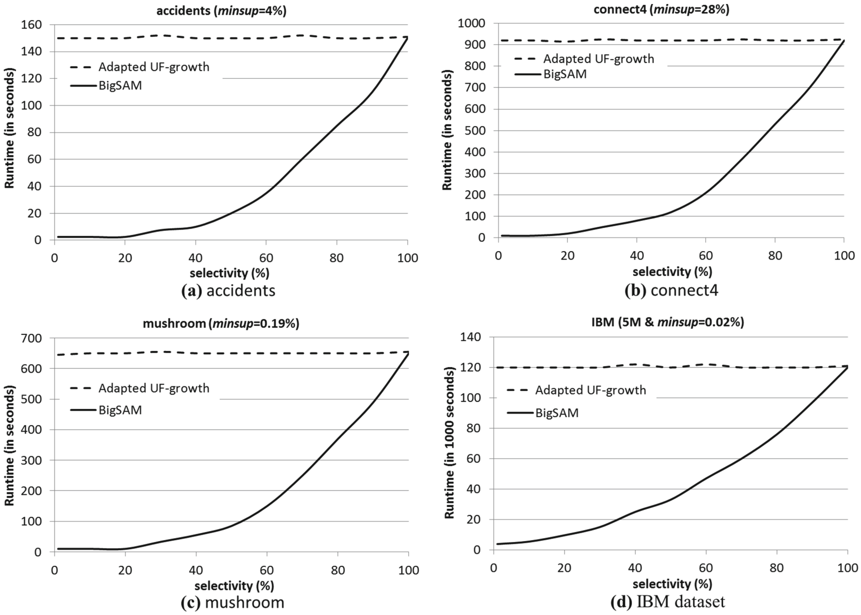
First, we used (i) a database consisting of items *all with existential probability value of 1* (indicating that all items are definitely present in the database and (ii) a user-specified SAM constraint. With this setting, we compared BigSAM (which finds valid frequent patterns from *uncertain* data) with CAP [21] (which also finds valid frequent patterns but from *precise* data). Experimental results showed that, (i) in terms of *accuracy*, BigSAM returned the *same* collection of valid frequent patterns as those returned by CAP. However, (ii) in terms of *functionality*, CAP is confined to finding valid frequent patterns from datasets when existential probability values of all items is 1, whereas our BigSAM algorithm is capable of finding valid frequent patterns from datasets containing items with *various existential probability values* ranging from 0 to 1.

Next, we used (i) a probabilistic database of uncertain data (containing items with various existential probability values ranging from 0 to 1) and (ii) a SAM constraint with 100 % selectivity (so that every item is selected). With this setting, we compared UF-growth [15] (which mines *unconstrained* frequent patterns from uncertain data) with BigSAM. Experimental results showed that, (i) in terms of *accuracy*, both algorithms returned the *same* collection of frequent patterns. (ii) In terms of *runtimes*, as shown in Fig. 1(a), both algorithms took shorter to run when *minsup* increased because fewer patterns had expected support ≥ *minsup*. (iii) Between the two algorithms, BigSAM (which was run in the MapReduce environment with 11 nodes) took much shorter runtimes than UF-growth [15] (which is a sequential algorithm). This led to a significant speedup,



(a) Runtime vs. *minsup* (#transactions = 5M) **(b)** Runtime vs. #transactions (*minsup* = 0.1%)

Fig. 1. UF-growth [15] vs. our BigSAM algorithm



(a) accidents **(b)** connect4 **(c)** mushroom **(d)** IBM dataset

Fig. 2. Constraint handling by our BigSAM algorithm

especially for mining Big data. (iv) As observed from Fig. 1(b), when the number of transactions increased, the gap between the runtimes of the two algorithms increased, and thus the speedup became more significant. (v) In terms of *functionality*, UF-growth [15] was not designed to handle constraints with selectivity other than 100 %, whereas our BigSAM algorithm is capable of handling constraints of *any selectivity*.

To adapt UF-growth for handling user-specified SAM constraints with selectivity other than 100 %, it first ignores the constraints and mines all frequent patterns, and then applies constraint checking as a post-processing step to check if each mined pattern is valid (i.e., satisfying the SAM constraints) and prune those uninteresting/invalid patterns (i.e., patterns that violate the SAM constraints). By doing so, the computation of this adapted algorithm is independent of the selectivity of the SAM constraints. Hence, it would be time- and space-consuming for handling uncertain Big data—especially when only a few frequent patterns are valid (i.e., low percentage selectivity). In contrast, Fig. 2 shows that (i) runtimes of BigSAM decreased when the selectivity was lower (i.e., when fewer frequent patterns were valid) on different datasets. Moreover, (ii) the runtime was proportional to the computation, which was proportional to the percentage selectivity. This illustrates the benefits of pushing constraint checking in our BigSAM algorithm.

As ongoing work, we are conducting more experiments to evaluate other aspects (e.g., the effect on the number of machines or cluster nodes, the communication costs) and with other algorithms (e.g., PUF-growth [16]).

5 Conclusions

In this paper, we proposed the BigSAM algorithm that (i) allows users to express their interest in terms of succinct anti-monotone (SAM) constraints and (ii) uses the MapReduce programming model to mine uncertain Big data for frequent patterns that satisfy user-specified constraints. As a result, our algorithm returns all and only those patterns that are interesting to the users. Experimental results show the effectiveness of our algorithm in mining these interesting patterns from probabilistic databases of uncertain Big data with MapReduce. As ongoing work, we are extending our algorithm for handling non-SAM constraints.

Acknowledgments. This project is partially supported by NSERC (Canada) and University of Manitoba.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: VLDB 1994, pp. 487–499 (1994)
2. Calders, T., Garboni, C., Goethals, B.: Efficient pattern mining of uncertain data with sampling. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010, Part I. LNCS, vol. 6118, pp. 480–487. Springer, Heidelberg (2010)
3. Chen, Y.-C., Ko, Y.-L., Peng, W.-C., Lee, W.-C.: Mining appliance usage patterns in smart home environment. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013, Part I. LNCS (LNAI), vol. 7818, pp. 99–110. Springer, Heidelberg (2013)
4. Chui, C.-K., Kao, B., Hung, E.: Mining frequent itemsets from uncertain data. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 47–58. Springer, Heidelberg (2007)
5. Condie, T., Mineiro, P., Polyzotis, N., Weimer, M.: Machine learning for Big data. In: ACM SIGMOD 2013, pp. 939–942 (2013)
6. Cordeiro, R.L.F., Traina, C., Traina, A.J.M., López, J., Kang, U., Faloutsos, C.: Clustering very large multi-dimensional datasets with MapReduce. In: ACM KDD 2011, pp. 690–698 (2011)

7. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. CACM **51**(1), 107–113 (2008)
8. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM SIGMOD 2000, pp. 1–12 (2000)
9. Koufakou, A., Secretan, J., Reeder, J., Cardona, K., Georgiopoulos, M.: Fast parallel outlier detection for categorical datasets using MapReduce. In: IEEE IJCNN 2008, pp. 3298–3304 (2008)
10. Kumar, A., Niu, F., Ré, C.: Hazy: making it easier to build and maintain big-data analytics. CACM **56**(3), 40–49 (2013)
11. Leung, C.K.-S.: Frequent itemset mining with constraints. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems, pp. 1179–1183. Springer, New York (2009)
12. Leung, C.K.S.: Big data mining and analytics. In: Wang, J. (ed.) Encyclopedia of Business Analytics and Optimization, pp. 328–337. IGI Global, Hershey (2014)
13. Leung, C.K.-S.: Mining uncertain data. WIREs Data Min. Knowl. Discov. **1**(4), 316–329 (2011)
14. Leung, C.K.-S., Hayduk, Y.: Mining frequent patterns from uncertain data with mapreduce for big data analytics. In: Meng, W., Feng, L., Bressan, S., Winiwarter, W., Song, W. (eds.) DASFAA 2013, Part I. LNCS, vol. 7825, pp. 440–455. Springer, Heidelberg (2013)
15. Leung, C.K.-S., Mateo, M.A.F., Brajczuk, D.A.: A tree-based approach for frequent pattern mining from uncertain data. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 653–661. Springer, Heidelberg (2008)
16. Leung, C.K.-S., Tanbeer, S.K.: PUF-Tree: a compact tree structure for frequent pattern mining of uncertain data. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013, Part I. LNCS (LNAI), vol. 7818, pp. 13–25. Springer, Heidelberg (2013)
17. Lin, M.-Y., Lee, P.-Y., Hsueh, S.-C.: Apriori-based frequent itemset mining algorithms on MapReduce. In: ICUIMC 2012, Article 76 (2012)
18. Luo, W., Chan, K.C.C.: Discovering patterns in drug-protein interactions based on their fingerprints. BMC Bioinform. **13**(S–9), S4 (2012)
19. MacKinnon, R.K., Leung, C.K.-S., Tanbeer, S.K.: A scalable data analytics algorithm for mining frequent patterns from uncertain data. In: Peng, W.-C., Wang, H., Bailey, J., Tseng, V.S., Ho, T.B., Zhou, Z.-H., Chen, A.L.P. (eds.) PAKDD 2014 Workshops, LNCS (LNAI), vol. 8643, pp. 404-416. Springer, Heidelberg (2014)
20. Madden, S.: From databases to big data. IEEE Internet Comput. **16**(3), 4–6 (2012)
21. Ng, R.T., Ng, Lakshmanan, L.V.S., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained associations rules. In: ACM SIGMOD 1998, pp. 13–24 (1998)
22. Riondato, M., DeBrabant, J.A., Fonseca, R., Upfal, E.: PARMA: a parallel randomized algorithm for approximate association rules mining in MapReduce. In: ACM CIKM 2012, pp. 85–94 (2012)
23. Tang, L.-Y., Hsiu, P.-C., Huang, J.-L., Chen, M.-S.: iLauncher: an intelligent launcher for mobile apps based on individual usage patterns. In: ACM SAC 2013, pp. 505–512 (2013)
24. Yang, S., Wang, B., Zhao, H., Wu, B.: Efficient dense structure mining using MapReduce. In: IEEE ICDM Workshops 2009, pp. 332–337 (2009)
25. Zaki, M.J.: Parallel and distributed association mining: a survey. IEEE Concurrency **7**(4), 14–25 (1999)

MBPD: Motif-Based Period Detection

Rasaq Otunba[1(✉)], Jessica Lin[1], and Pavel Senin[2]

[1] George Mason University, Fairfax, VA 22030, USA
{rotunba, jessica}@gmu.edu
[2] University of Hawaii, Honolulu, HI 96822, USA
senin@hawaii.edu

Abstract. Massive amounts of data are generated daily at a rapid rate. As a result, the world is faced with unprecedented challenges and opportunities on managing the ever-growing data. These challenges are prevalent in time series for obvious reasons. Clearly, there is an urgent need for efficient solutions to mine large-scale time series databases. One of such data mining tasks is periodicity mining. Efficient and effective periodicity mining techniques in big data would be useful in cases such as finding animal migration patterns, analysis of stock market data for periodicity, and outlier detection in electrocardiogram (ECG), analyses of periodic disease outbreak etc. This work utilizes the notion of time series motifs for approximate period detection. Specifically, we present a novel and simple method to detect periods on time series data based on recurrent patterns. Our approach is effective, noise-resilient, and efficient. Experimental results show that our approach is superior compared to a popularly used period detection technique with respect to accuracy while requiring much less time and space.

Keywords: Motif · Period · Time series · Big data

1 Introduction

Periodicity is the tendency of a pattern to recur at regular intervals, which are referred to as periods. Periodic patterns occur in many natural phenomena or human activities. Examples include an employee's daily work schedule, yearly migration pattern of animals, regional sunspot cycle etc. These data can be so large and complex that it becomes difficult to process using traditional database management tools or data processing applications. Such data is typically referred to as big data. The fact that we are in the era of big data cannot be overemphasized especially with the large amount of available data from the internet and ubiquitous computing devices that are now parts of our everyday lives. Detecting the period in data, big or "*not so big*", can provide useful insight on the data, help make better predictions, detect anomalies and improve similarity matching [16] among other things. It's imperative to point out that the focus of this work being time series makes it suitable for other kinds of data such as multimedia because they can be converted to time series e.g. the extraction of MFCC from audio as it is used for one of the datasets in our experiments. Several methods have been proposed to detect periods in data. Most of the existing methods are particularly suitable for perfect periods, which is hardly the case in most natural phenomena. While the related problem of finding the exact period of a time series is a simpler one to solve,

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 793–804, 2014.
DOI: 10.1007/978-3-319-13186-3_71

it may be too restrictive for real-world phenomena. Periods in real datasets are typically noisy and incomplete. That is, while the periodic patterns exhibit tangible similarity, they may not always be identical and equally distributed. These factors warrant robust approximate period detection schemes like our solution and it has even been shown that in many applications, approximate solutions are sufficient [30]. A robust solution should also be able to detect periods in an efficient manner. In general, three types of periodic patterns can be detected in a time series as illustrated by Rasheed et al. [1] and they are described as follows:

- a time series exhibits partial periodicity if at least one symbol in addition to at least one variable symbol is periodic. For instance, in time series T = wxyz wxxy wxyy wxwz, the sequence wx is periodic with period p = 4; and the partial periodic pattern wx ** exists in T, where * denotes a variable symbol.
- a time series exhibits symbol periodicity if at most one symbol is repeated periodically. For example, in time series T = xyz xzy xxy xyy, symbol x is periodic with period p = 3. We consider this to be a special case of partial periodicity when the periodic subsequence has one symbol and argue that a technique that can detect subsequence periodicity can detect symbol periodicity.
- a time series exhibits segment periodicity if an entire pattern is periodic. For instance, the time series T = wxyz wxyz wxyz wxyz has a segment period p = 4. The periodic segment is wxyz.

Most techniques are suitable for discrete sequences. However, time series are real-valued data. To adapt the periodicity definitions described above, we need a pre-processing step that discretizes the real-valued time series into a symbolic representation. The standard pre-processing approach is to use SAX (Symbolic Aggregate approXimation) [2], a well-known discretization technique, to convert a time series into a string or a set of strings [3]. In this work, we propose a novel technique to detect periods in time series data by learning the repeated patterns (motifs) from data. To the best of our knowledge, this work is the first to use motif discovery as a means to period detection in time series.

A time series motif is a pattern that consists of two or more similar subsequences based on some distance threshold [3]. While our approach can work with any motif discovery algorithms, in this work we focus on a recently proposed variable-length motif algorithm based on grammar induction called GrammarViz[1] [3]. GrammarViz consists of two major steps: (1) extracting subsequences via a sliding window converting them to strings via SAX; and (2) infers a set of context-free grammar rules on the sequence of strings using Sequitur [4]. The grammar rules represent repeated patterns in the sequence, each of which can be regarded a time series motif. GrammarViz is an ideal basis for our work because of its simplicity, space- and time-efficiency, and most importantly, its ability to detect variable-length motifs, as these properties are transferred to our periodicity detection method.

[1] Although not explicitly named in the paper, the authors refer to it as GrammarViz on their website: http://www.cs.gmu.edu/∼jessica/GrammarViz.html.

In summary, we propose a simple and elegant approximate periodicity detection algorithm, Motif-Based Period Detection (MBPD). Our work makes the following novel contributions:

- We propose to use time series motif discovery on the string representation as an antecedent to approximate periodicity detection on the original time series.
- Our algorithm is both time- and space-efficient, and is suitable for streaming data.
- We introduce a simple ranking method for the most significant period.
- We extended GrammarViz and implemented the periodicity visualization feature that allows users to navigate and sort the detected periods.
- We conducted experiments to compare the performance of our technique against other popular techniques on synthetic and real datasets.

The rest of the paper is organized as follows. Section 2 discusses related work while Sect. 3 outlines preliminaries. We describe our approach in Sect. 4. Section 5 describes the experiments performed. We conclude with limitations of our approach and make recommendations for improvement as future work in Sect. 6.

2 Related Work

Existing periodicity detection methods can be categorized based on a number of factors [1, 5–14, 16, 24, 25]. These factors include parameter dependency, the type of periodicity detected, the span of periodicity detected and the domain in which the periodicity is detected. Some methods require the specification of the period value [12–14]. This is not ideal as the detection of the period value is in itself a task worthy of due consideration. Some of these methods detect only symbol periodicity [8, 26]. Yang et al. [9, 24] proposed a linear time distance-based technique for discovering the potential periods in a time series. However, their method fails to detect some valid periods because only adjacent intervals are considered. Rasheed et al. [1] proposed an algorithm to detect periodicity in time series using suffix trees. The time requirement for their proposed method can rise to the order of $O(n^3)$. Most algorithms detect only a subset of the types of periodicity (symbol, sequence or segment) mentioned earlier. The method proposed by Han et al. in [7] detects only segment periodicity. Certain techniques [12, 13], which are based on another technique, ParPer [14], are suitable for detecting sequence periodicity in time series. ParPer makes use of peculiar properties e.g. apriori property related to sequence periodicity in a time series for periodicity detection. Most of the aforementioned techniques suffer from noise sensitivity. WARP [11] was developed to be noise resilient but it detects only segment periodicity. Few techniques [12, 14] detect subsection periodicity while most are meant for full-cycle periodicity detection.

Periodicity detection algorithms can also be classified into time domain and frequency domain methods. Time domain methods are based on autocorrelation functions while frequency domain methods are based on spectral density functions. The premise for using time domain methods is that the autocorrelation function of a periodic data has the same period as the data with peaks obtained at time $t = 0$, period T, and multiples of T. Time domain methods are suitable for sinusoidal signals and they are not noise resilient. Frequency domain methods, on the other hand, decompose signals into constituent frequency components. The result of frequency domain methods is a power spectral density with impulses determined by the corresponding Fourier coefficients. These Fourier coefficients can be extracted to create a periodogram [17].

Autocorrelation and Fourier Transforms are two of the most popular periodicity detection techniques [15]. Autocorrelation is able to detect short and long periods, but creates difficulty in identifying the true period due to the fact that the multiples of the true period will have the same power as the true period. On the other hand, Fourier transforms suffer from a number of problems: spectral leakage, which causes a lot of false positives in the periodogram, and poor estimation of long periods due to issues with low frequency regions or sparseness in data [18]. Some methods combine both autocorrelation and Fourier transforms [6, 16].

Our method is able to detect the most significant period in a dataset without requiring the period value as a parameter and this is done in the time domain. Our method also detects the different types of periodicity.

3 Preliminaries

In this section we define periodicity, approximate periodicity and the problem addressed in this work.

Definition 1. Let $S = t_0,\ t_1...t_{n-1}$ be a string representation of a time series with length of n, i.e. $|S| = n$. S is said to be periodic if $S(t) = S(t + p)$, where $t \in N,\ t \geq 0,\ t < n - p$, T is a subsequence of S such that, $T = t_0,\ t_1...t_{p-1},\ |T| = p,\ p \geq 1$ and $p \leq n/2$. The smallest such subsequence T is called the period of S. If no such period T can be found in S, S is said to be aperiodic. For example, if S = wxywxywxy, the period T = wxy, p = 3.

For clarity, we do not consider the substring of length (m*p) to be a period of S for any m such that $m \neq 1$.

Definition 2. Let S be n-long string over alphabet Σ. Let r be an error function defined on strings. S is called periodic with k error on T if there exists a string T over Σ, such that r(S, T) = k i.e. r evaluate the error of assuming T is the period of S. The string T that evaluates to smallest such k is called the approximate period of S. For example, let S = wxywxywxy, r(S, A) ≥ r(S, B) where A = wxy occurring at positions 0, 3 and 6, and B = wxy occurring at positions 0 and 6, A and B are candidate periods, A is the approximate period of S otherwise referred to as the most significant period of S.

Problem Definition. Given a string function r, and String S of length n over alphabet compute the approximate period T under the function r.

3.1 Motif Discovery

Time series motifs are repeated similar patterns. We argue that detecting the frequent patterns could serve as an antecedent to periodicity detection. Many algorithms have been proposed to find motifs in time series data [3, 19–22]. In this work, we focus on GrammarViz, a fast, approximate variable-length motif discovery algorithm based on grammar induction [3]. The factors we considered in choosing a motif discovery method for this work include efficiency with respect to space and time, the ability to detect the motifs in a streaming fashion, the ability to detect variable length motifs, the ability to detect periodicity in string representation of time series, and simplicity. GrammarViz utilizes Sequitur [4], a context-free grammar induction technique, to derive rules considered to be motifs from string representation of time series. These motifs are mapped back to the original time series to show their occurrences. A benefit derived from the ability to work on string representation is the fact that the technique is applicable to many kinds of data whose dimensionality can be reduced by discretization to string symbols. GrammarViz is the first time series motif discovery algorithm that can detect variable-length motifs in an effective and efficient manner, and it is able to do so in a streaming fashion. The authors of GrammarViz also created a visualization tool for clarity and easy navigation of the produced results. GrammarViz achieves variable-length motif discovery as a result of numerosity reduction, thus making MBPD suitable for cases where a periodic pattern may occur with variable lengths in a time series.

4 Our Approach

The fundament premise of our approach is to first discover the motifs in the time series with high efficiency and effectiveness and then detect the most periodic motif.

Algorithm 1. Motif-Based Period Detection

```
INPUT:  String S of length n over alphabet $\Sigma$.
OUTPUT:  The approximate period of S, T.

1.  /* Find the rule/motif objects M = {$m_1$, $m_2$, . . . $m_d$} from GrammarViz algorithm */
2.  M = grammarViz(S);
3.  /* Compute periods and errors for each motif, return the one with the smallest error*/
4.  $m_1$ = periodicity($m_1$);
5.  $p_1$ = $m_1$.getPeriod();
6.  $r_1$ = $m_1$.getError();
7.  rMin = $r_1$  // store the minimum error in rMin
8.  for each $m_i \in$ M do
9.      $m_i$ = periodicity($m_i$);
10.     $p_i$ = $m_i$.getPeriod();
11.     $r_i$ = $m_i$.getError();
12.     if ($r_i$ < rMin)
13.         approxP = $p_i$;
14.         rMin = $r_i$;
15. end for
16. return approxP;
```

Algorithm 1 shows the pseudocode for the MBPD algorithm. The motif objects returned in Line 2 are stored along with the start and stop positions of each occurrence in the time series. We consider only periods that occur at least 3 times in a time series for this work since anything less has a higher probability of being a false positive but it's trivial to modify the algorithm to detect periods that occur twice if desired. Lines 3–7 and the loop from Lines 8–14 computes the period of each motif, the error (our r function from Definition 2) defined by the standard deviation of the intervals of all occurrences and the approximate period. The period of each motif is calculated as the mean of intervals (between the start positions of two consecutive occurrences) of all occurrences in the time series. Both computations of the approximate period and error are done on the original time series after the derived string motifs are mapped back to the original time series. The approximate period of the time series is the period corresponding to the lowest error. The periodicity function called on Lines 4 and 9 of Algorithm 1 is shown in Algorithm 2.

The efficiency of MBPD largely depends on the efficiency of GrammarViz (Line 2), which has Sequitur at its core. GrammarViz has linear time and space complexity. As a result, the time complexity of MBPD is O(n*k) for a time series of size n, where k is the average number of instances for each motif rule produced by Sequitur. The space complexity is still O(n) because the memory space needed for variables used in Algorithms 1 and 2 are negligible. Compared to most existing methods for time series periodicity detection, MBPD has a competitive space and time complexity.

Algorithm 2. Periodicity Algorithm

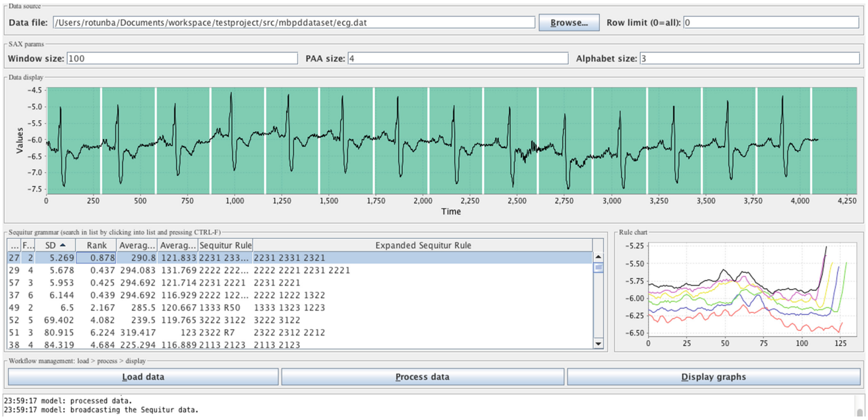
```
INPUT:  Motif M with start positions A = {$a_1, a_2, . . . a_b$} for all b occurrences
OUTPUT:  Motif M with the period and error set respectively

sum_Interval = 0, sqd = 0;
for each $a_i \in$ A do
    sum_Intervals = sum_Intervals + $a_i$ - $a_{i-1}$;
end for
M.period = sum_Intervals/(b-1);
for each $a_i \in$ A do
    sqd = sqd + (($a_i$ - $a_{i-1}$ - M.period) ^ 2);
end for
M.error = (sqd/(b-1)) ^ 0.5;
return M;
```

5 Experiment

In this section we evaluate MBPD on synthetic, pseudo-synthetic and real datasets. Our periodicity detection and visualization software is an extension of GrammarViz. More details about the visualization tool and the GrammarViz algorithm in line 2 of Algorithm 1 can be found in [3]. Experiments were performed on a 2.7 GHz, Intel Core i7, MAC OS X version 10.7.5 with 8 GB memory.

Figure 1 is a snapshot of the visualization tool showing the approximate periodicity detected in an ECG dataset in the data display section of the figure. Other periods can be viewed by navigating the list of rules in the sequitur grammar section of the figure.

We compare our method with Fast Fourier Transform (FFT). The frequency with the highest spectral power from FFT of the dataset is converted into time domain and considered as the most significant period. FFT is chosen for its suitability for real-valued datasets.



Data source

Data file: /Users/rotunba/Documents/workspace/testproject/src/mbpddataset/ecg.dat

Row limit (0=all): 0

SAX params

Window size: 100

PAA size: 4

Alphabet size: 3

Sequitur grammar (search in list by clicking into list and pressing CTRL-F)

...	F...	SD ▲	Rank	Averag...	Averag...	Sequitur Rule	Expanded Sequitur Rule
27	2	5.269	0.878	290.8	121.833	2231 233...	2231 2331 2321
29	4	5.678	0.437	294.083	131.769	2222 222...	2222 2221 2231 2221
57	3	5.953	0.425	294.692	121.714	2231 2221	2231 2221
37	6	6.144	0.439	294.692	116.929	2222 122...	2222 1222 1322
49	2	6.5	2.167	285.5	120.667	1333 R50	1333 1323 1223
52	5	69.402	4.082	239.5	119.765	3222 3122	3222 3122
51	3	80.915	6.224	319.417	123	2322 R7	2322 2312 2212
38	4	84.319	4.684	225.294	116.889	2113 2123	2113 2123

```
23:59:17 model: processed data.
23:59:17 model: broadcasting the Sequitur data.
```

Fig. 1. Snapshot of MBPD visualization tool detecting periodicity in ECG dataset.

It is worth mentioning that while we considered other state-of-the-art techniques such as STNR [1], WARP [11], and the probability-based method in [18] for comparison, we found that they are not suitable for our purpose for the following reasons other than their unsuitableness for real values. WARP caused an out of memory exception for the large datasets (65636 data points) used in our experiments and returned unintuitive results for most of the other datasets e.g. 515 (43 years) for the Zürich sunspot dataset whereas the proper period for the dataset is 132 (11 years). The out of memory exception is most probably due to WARP's O(m^2) space complexity for a time series of size m.; STNR on the other hand returns many candidate periods even with the pruning techniques suggested, which deviates from our goal of finding the most significant period. Finally the method proposed in [18] is meant for binary sequences representation of Boolean-type observations and not real-valued sequences.

5.1 Datasets

We used 12 datasets of various periodicity, noises, and lengths in our experimental evaluation. We ensured the length of each dataset is a power of 2 to avoid introducing bias by padding the dataset with zeros in order to use FFT for comparison. A subplot of all 12 datasets is shown in Fig. 2. Figures 3 and 4 show the periodicity detected in 2 of the datasets used in our experiments.

Synthetic Datasets. We created 6 synthetic datasets with various properties that could affect performance.

- S_ONE: 65636 points to depict perfect segment periodicity by repeating 10000 points 7 times except for the last 4364 points.
- S_TWO: 65636 points to depict perfect segment periodicity by repeating 10000 points 7 times and except for the last 4364 points.
- S_THREE: 65636 points to demonstrate sensitivity to noise by introducing noise in the form of insertion, deletion and replacement into S_ONE, 10 % each.

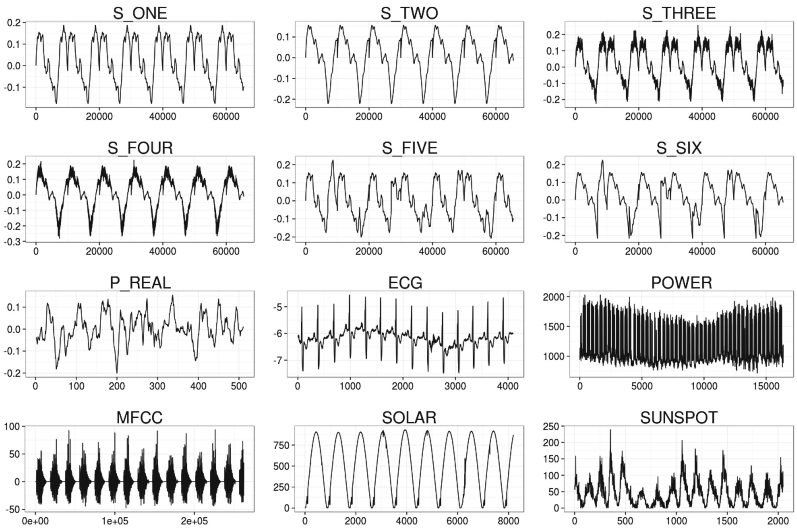


Fig. 2. A subplot of all 12 datasets used for experiments

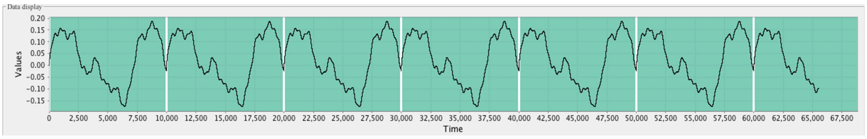


Fig. 3. Snapshot of MBPD visualization tool detecting periodicity in S_ONE dataset.

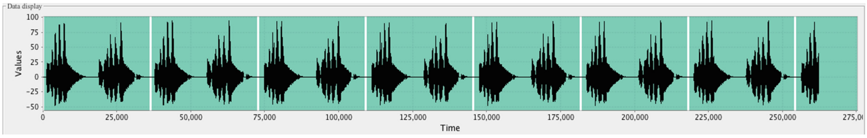


Fig. 4. Snapshot of MBPD visualization tool detecting periodicity in MFCC dataset.

- S_FOUR: 65636 points to demonstrate sensitivity to noise by introducing noise in the form of insertion, deletion and replacement into S_TWO, 10 % each.
- S_FIVE: 65636 points to depict subsequence periodicity by repeating a portion of the repeated 10000 points in S_ONE and leaving the remaining as variable points.
- S_SIX: 65636 points to depict subsequence periodicity by repeating a portion of the repeated 10000 points in S_TWO and leaving the remaining as variable points.

Real Datasets. We used 5 real datasets.

- ECG: 4096 points of ECG (electrocardiogram) data
- POWER: 16384 points of power consumption data

- MFCC: 262144 points of MFCC (Mel-Frequency Cepstral Coefficients) extracted from Rufous-collared Sparrow bird song which can be found at http://www.xeno-canto.org/120810
- SOLAR: 8192 points of solar data
- SUNSPOT: 2048 points of sunspots on Zürich.

Pseudo-Real Dataset (P_REAL). This dataset has 512 points. We use this dataset to depict periodicity detection in a short dataset. It's a sea surface temperature dataset for North Atlantic Ocean (simulated data for 1000 years) with 10-year moving average smoothing.

5.2 Results

We evaluated the performance of our method against FFT with respect to the ranking error rates on both the synthetic and real datasets as shown in Table 1. In Table 2, we show the error rates of the period value detected on the synthetic datasets.

Ranking error rate is computed by dividing the rank position, *i* (starting from zero) of the most significant period by the size of the time series e.g. if the candidate periods returned by MBPD are 3, 4, 7 and 9 in that order when ranked and 4 is the most significant period, the ranking error rate of MBPD for this dataset is ¼ (0.2500). Error rate of the period values is computed as follows in Eq. 1:

$$Error\ rate\ of\ period = \frac{|\boldsymbol{Expected\ value - Actualvalue}|}{\boldsymbol{Actual\ value}} \tag{1}$$

Since we do not know the exact periods in the real datasets, we did not evaluate the error rate of the period values. Table 3 contains the periods detected on the real datasets as well as the expected range of values. As shown in Tables 1 and 2, MBPD ranks the most significant period better and detects the period more accurately than FFT. Since we are concerned with the most significant period, an improper ranking otherwise

Table 1. Ranking error rate on synthetic and real datasets.

Datasets	MBPD	FFT
S_ONE	0.0000	0.0000
S_TWO	0.0000	0.0000
S_THREE	0.0000	0.0000
S_FOUR	0.0000	0.0000
S_FIVE	0.0000	0.0000
S_SIX	0.0000	0.0000
P_REAL	0.0000	0.0000
ECG	0.0000	0.0002
POWER	0.0000	0.0001
MFCC	0.0000	–
SOLAR	0.0000	0.0001
SUNSPOT	0.0000	0.0005

Table 2. Period error rate on synthetic datasets.

Datasets	MBPD	FFT
S_ONE	0.0000	0.0637
S_TWO	0.0000	0.0637
S_THREE	0.0009	0.0637
S_FOUR	0.0011	0.0637
S_FIVE	0.0000	0.0637
S_SIX	0.0000	0.0923

Table 3. Period values on real datasets.

Datasets	MBPD	FFT	EXPECTED VALUES
P_REAL	77.00	85.33	75–85
ECG	290.80	292.57	290–295
SOLAR	870.60	910.22	870–875
MFCC	36340	–	36000–36500
POWER	328.98	334.37	325–330
SUNSPOT	135.85	136.53	132–137

referred to as false dismissal experienced by using FFT is undesirable. Even though we do not know the exact period for the real datasets, the visualization tool helps by allowing us to visualize the results, e.g. the highlighted patterns in Figs. 1, 3, and 4.

For some of the real datasets, we have some prior knowledge on what to expect for the periodicity. The Zürich sunspot data, for example, is known to have a period of about 11 years (132 months) as described in [28]. All three techniques produced reasonable approximations for the dataset. The MFCC data is extracted from a bird song, which has a period between 30 K–40 K when visualized in the software and listened to meticulously. As seen in Table 3, all 3 techniques performed competitively on 4 of the real datasets but only MBPD detected the period in the MFCC extraction. This also makes MBPD stand out as a superior technique. We did not record the period and ranking error rate for FFT on the MFCC dataset because the 10 most significant periods (3.33, 4.00, 3.33, 4.00, 2.86, 2.86, 2.86, 3.33, 3.34, 4.00) were spurious altogether and we don't know which of them should be selected for evaluation as they are all far from the expected period. We consider only the first 10 coefficients in this work because the first 10 coefficients are known to contain approximately 90 % of the energy [29]. We attribute the poor result of FFT to the known issue of FFT regarding the low frequency regions which translates to issues in detecting long periods as is the case with the MFCC dataset (35000–36000).

6 Conclusion and Future Work

We present an approximate periodicity detection scheme in this work. We evaluated our approach against popular techniques on synthetic and real datasets. Our technique is highly competitive with respect to efficiency and effectiveness as well as being robust to noise. We also utilized a visualization tool for this work. Even though the intention is to detect the most significant approximate period in the dataset, our visualization tool permits the navigation of other periods. As future work, we would like to extend the work to detect the exact or at least approximate span of the periods detected in addition to detecting the periodic pattern with high confidence. As this work seeks to motivate the use of motif discovery as an antecedent to periodicity detection, we do not claim that GrammarViz is the best choice of algorithm for motif discovery. Since GrammarViz is an approximate motif discovery algorithm, it may not find all of the motifs, which in turn may impact the quality of periods detected by our algorithm. We believe that using a better grammar induction algorithm or, more generally, a more aggressive motif discovery technique as an antecedent could enhance the performance of MBPD. Nevertheless, the benefit of finding variable-length patterns and the ability to do so efficiently as permitted by GrammarViz is highly desirable and beneficial to our algorithm.

References

1. Rasheed, F., Al-Shalalfa, M., Alhajj, R.: Efficient periodicity mining in time series databases using suffix trees. In: TKDE (2011)
2. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Workshop on Research Issues in DMKD (2003)
3. Li, Y., Lin, J., Oates, T.: Visualizing variable-length time series motifs. In: SDM (2012)
4. Nevill-Manning, C.G., Witten, I.H.: Identifying hierarchical structure in sequences: a linear-time algorithm. J. Artif. Intell. Res. **7**, 67–82 (1997)
5. Amir, A., Eisenberg, E., Levy, A.: Approximate periodicity. In: Cheong, O., Chwa, K.-Y., Park, K. (eds.) ISAAC 2010, Part I. LNCS, vol. 6506, pp. 25–36. Springer, Heidelberg (2010)
6. Berberidis, C., Aref, W., Atallah, M., Vlahavas, I., Elmagarmid, A.: Multiple and partial periodicity mining in time series databases. In: ECAI (2002)
7. Han, J., Gong, W., Yin, Y.: Mining segment-wise periodic patterns in time related databases. In: KDD (1998)
8. Ma, S., Hellerstein, J.: Mining partially periodic event patterns with unknown periods. In: ICDE (2001)
9. Yang, J., Wang, W., Yu, P.: Mining partial periodic patterns with gap penalties. In: ICD (2002)
10. Elfeky, M.G., Aref, W.G., Elmagarmid, A.K.: Periodicity detection in time series databases. In: ICDE (2005)
11. Elfeky, M.G., Aref, W.G., Elmagarmid, A.K.: WARP: time warping for periodicity detection. In: ICDM (2005)
12. Sheng, C., Hsu, W., Lee, M.L.: Mining dense periodic patterns in time series data. In: ICDE (2006)
13. Sheng, C., Hsu, W., Lee, M.L.: Efficient Mining of Dense Periodic Patterns in Time Series. Technical report, Nat'l Univ. of Singapore (2005)
14. Han, J., Yin, Y., Dong, G.: Efficient mining of partial periodic patterns in time series database. In: ICDE (1999)

15. Priestley, M.B.: Spectral Analysis and Time Series. Academic Press, London (1981)
16. Vlachos, M., Yu, P.S., Castelli,V.: On periodicity detection and structural periodic similarity. In: SDM (2005)
17. Stoica, P., Moses, R.L.: Introduction to Spectral Analysis. Prentice-Hall, Upper Saddle River (1997)
18. Li, Z., Wang, J., Han, J.: Mining event periodicity from incomplete observations. In: KDD (2012)
19. Lam, H.T., Pham, N.D., Calders, T.: Online discovery of top-k similar motifs in time series data. In: SIAM Conference on Data Mining, SDM (2011)
20. Lin, J., Keogh, E., Lonardi, S., Patel, P.: Finding motifs in time series. In: Proceedings of 2nd Workshop on Temporal Data Mining at KDD (2002)
21. Mueen, A., Keogh, E.J.: Online discovery and maintenance of time series motifs. In: KDD (2010)
22. Nunthanid, P., Niennattrakul, V., Ratanamahatana, C.: Discovery of variable length time series motif. In: ECTICON (2011)
23. Smyth, W.F.: Computing periodicities in strings — a new approach. In: Proceedings of the 16th Australasian Workshop on Combinatorial Algorithms (2007)
24. Yang, J., Wang, W., Yu, P.S.: Mining asynchronous periodic patterns in time series data. In: KDD (2000)
25. Elfeky, M.G., Aref, W.G., Elmagarmid, A.K.: Using convolution to mine obscure periodic patterns in one pass. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 605–620. Springer, Heidelberg (2004)
26. Scargle, J.D.: Studies in astronomical time series analysis. II - statistical aspects of spectral analysis of unevenly spaced data. In. Astrophys. J. **263**, 835–853 (1982)
27. https://jmotif.googlecode.com
28. http://solarscience.msfc.nasa.gov/SunspotCycle.shtml
29. Vlachos, M.: A practical time-series tutorial with matlab. In: PKDD (2005)
30. Arora, S.: Approximation schemes for np-hard geometric optimization problems: a survey. Math. Prog. **97**, 43–69 (2003)

A Model-Based Multivariate Time Series Clustering Algorithm

Pei-Yuan Zhou(✉) and Keith C.C. Chan

Department of Computing, The Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong
choupeiyuan@gmail.com, cskcchan@comp.polyu.edu.hk

Abstract. Given a set of multivariate time series, the problem of clustering such data is concerned with the discovering of inherent groupings of the data according to how similar or dissimilar the time series are to each other. Existing time series clustering algorithms can divide into three types, raw-based, feature-based and model-based. In this paper, a model-based multivariate time series clustering algorithm is proposed and its tasks in several steps: (i)data transformation, (ii)discovering time series temporal patterns using confidence value to represent the relationship between different variables, (iii) clustering of multivariate time series based on the degree of patterns discovering in (ii). For evaluate performance of proposed algorithm, the proposed algorithm is tested with both synthetic data and real data. The result shows that it can be promising algorithm for multivariate time series clustering.

Keywords: Multivariate time series · Model-based clustering algorithm · Time series clustering

1 Introduction

A *multivariate time series* can be considered as made up of a collection of data values taken by a set of temporally interrelated variables monitored over a period of time at successive time instants spaced at uniform time intervals. Multivariate time series data are commonplace in business and finance, social and biological sciences, engineering and computing, medicine and healthcare, etc., and effective clustering of such data can have many applications in many problem domains. Time series clustering has received a lot of attention in the past and a number of different approaches have been developed to tackle the problem [3–5, 12, 14, 18]. These approaches can be classified into two types depending on whether raw data are used directly or indirectly in the clustering process [1]. And time series clustering algorithms that do not use raw-data directly can be classified into two types depending on whether they take a *feature-based* or a *model-based approach* [2]. The feature-based approaches are best used when relevant knowledge about a problem domain is available [6]. Otherwise, the model-based approaches to time series clustering can be considered.

Algorithms that adopt the model-based approaches assume that each time series can be approximated by a known mathematical model defined by a set of parameters that can be accurate estimated [2]. They also assume that time series that are similar to each

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 805–817, 2014.
DOI: 10.1007/978-3-319-13186-3_72

other can be approximated by similar models. Under such assumptions, the similarity between two time series can be approximated by the similarity between their corresponding models. Many time series clustering algorithms described above have been shown to be very effective at their tasks. However, these algorithms are mainly developed to handle univariate time series. How they can be modified to deal with multivariate time series data is not obvious. For a time series clustering algorithm to be useful for more complex tasks, it has to be able to tackle multivariate time series.

In this paper, we develop a proposed algorithm which is a model-based clustering algorithm but it is application-independent and can perform its tasks even without any domain knowledge about relevant features or any assumption about underlying data models. It can handle both continuous and discrete data and data of relatively high noise contents. Given a set of multivariate time series each of which consists of a set of component univariate time series, the proposed algorithm discovers clusters in the data by discovering temporal patterns in each multivariate time series and compare them with those discovered in the others so that multivariate time series that exhibit similar patterns can be grouped together in the same cluster. To discover and make use of these patterns for clustering, the algorithm performs several steps (i) data transformation, (ii) discovering time series temporal patterns using confidence value to represent the relationship between different variables, (iii) clustering of multivariate time series based on the degree of patterns discovering in (ii).

The structure of this paper is arranged like following: in Sect. 2, we present a summary of existing work on time series clustering and discuss the different kind of problems that they can be best used to tackle. In Sect. 3, we describe the details of each step that proposed algorithm takes when performing its tasks. The algorithm has been evaluated with both simulated and real data sets and the details of how the performance of it is tested are presented in Sect. 4. In this same section, we also discuss results of the various tests we carried out evaluations for effectiveness of its tasks. Finally, in the last section, we present a summary of the paper and discuss possible directions for future work.

2 Related Work

2.1 Multivariate Time Series

Existing algorithms for time series clustering can be classified into two types, those that take the *direct approach* and those that take the *indirect* approach [2]. Clustering algorithms that take the direct approach work directly with raw time series data in time or frequency domain and they are sometimes referred to as the *raw-data-based approaches* [2]. Many raw-data-based approaches to time series clustering are proposed mainly to handle univariate time series [2]. The algorithm proposed in [18] is one of the few exceptions that is able to cluster non-stationary time series using a locally stationary version of the *Kullback-Leibler discrimination information measure*. The algorithm can be used to handle multivariate time series of equal length but it is not for use with mixed continuous and discrete time series data.

The indirect approaches to time series clustering can be classified into the category of *feature-based approaches* [2] (a.k.a. *representation-based approaches* [8]) and the category of *model-based approaches* respectively [2]. Time-series clustering algorithms that are classified as the *feature-based* approaches [2] perform their tasks with features extracted from the raw data rather than with the raw data directly. The advantage with the use of such clustering algorithms is that they can now work with lower dimensional space and this is especially important when data are collected at fast sampling rates. An example of a feature-based clustering algorithm can be found in [5] where an algorithm for detecting activated voxels in event-related BOLD fMRI data is proposed. Another example of a feature-based approach to time series clustering is given in [12]. The raw time series data is first converted into feature vectors of lower dimension using an Independent Component Analysis (ICA) algorithm. Other than the ICA, other feature-based approaches that also make use of popular dimension reduction methods such as the *Principal Component Analysis* (PCA) have also been proposed [8, 9]. As traditional PCA techniques cannot be used with multivariate time series data, there have been some attempts to overcome this limitation to use a Generalized Principal Component Analysis (GPCA) technique to make PCA usable with multivariate time series data [14].

2.2 Model-Based Approach

Besides that basic information about multivariate time series clustering algorithm, the model-based approach need to be specified. Model-based approaches to time series clustering assume that each time series is generated by some known models. For example, the autoregressive (AR) model is used in [15] to deal with univariate ARIMA time series data. Under the assumption of such model, the *k*-medoids algorithm is then used, together with the Euclidean distance as a measure of dis-similarity between the Linear Predictive Coding (LPC) cepstra of two time-series in the clustering process. In [17], a model-based approach for clustering univariate ARIMA time series was also proposed under the assumption that the time series data are generated by *k* different ARMA models, with each model corresponds to one cluster of time series.

Other than the ARMA model, which is rather popularly used in different areas of applications, another model-based approach that has been considered is to use a framework based on discrete HMMs. In [10], an algorithm is proposed for “tool wear monitoring” in a machining process. The algorithm makes use of feature vectors extracted by wavelet analysis of vibration signals. The signal data are then converted into a symbol sequence by vector quantization, which in turn is used as input for training the HMM by an expectation maximization approach. Other than the work as described in [10], a Bayesian HMM clustering algorithm has also been proposed in [9] for ecology data. It uses the Bayesian Information Criterion (BIC) in a sequential search strategy to select between models. The sizes of the individual HMMs are dynamically determined for each cluster, so the strategy starts with the simplest model, gradually increases the model size, and stops when the BIC score of the current model is less than that of the previous model.

3 Details of Proposed Algorithm

With the above requirements in mind, we have developed an algorithm for clustering multivariate time series. Given a set of multivariate time series (MVTS), the algorithm discovers clusters in the data by discovering temporal patterns in each MVTS. The patterns discovered in one MVTS are compared against those discovered in the others so that MVTS that have similar discovered temporal patterns are grouped together into the same cluster.

For discovering temporal patterns, it should be noted that, as each MVTS is made up of data obtained by monitoring a set of temporally interrelated variables over a period of time, the variables do not take on random values. Instead, at any time instant, these variables can take on values that may be temporally related to its previous values or to previous values of the other variables. One main task of MUTSCA is, therefore, to uncover these temporal interrelationships. As each of the variables being monitored generates a component univariate time series (CUVTS) of a MVTS, the main task of MUTSCA is, in other words, to uncover the temporal interrelationships between values observed at different time instants within a CUVTS or between two or more CUVTS. These temporal interrelationships constituent what can be called intra-CUVTS and inter-CUVTS temporal patterns respectively in each MVTS. MUTSCA, hence, performs its tasks in several steps (i) data transformation, (ii) discovering intra-CUVTS temporal patterns within each MVTS, (iii) discovering inter-CUVTS temporal patterns within each MVTS; (iv) clustering of MVTS based on the similarities and dissimilarities of the intra- and inter-CUVTS temporal patterns. In the following sections, we present details of each of these steps.

3.1 The Problem and the Notations

Let **S** represent a set of multivariate time series data with the following characteristics:

1. **S** consists of N MVTS represented as $\mathbf{S} = \{\mathbf{S}^1, \mathbf{S}^2, \ldots, \mathbf{S}^N\}$.
2. Each of the $\mathbf{S}^i, i = \{1, \ldots, N\}$ consists of n CUVTS that can be represented as S_j^i, j = 1, …., n, respectively, so that $\mathbf{S}^i = \{S_1^i, S_2^i, \ldots, S_n^i\}$.
3. Each of the n CUVTS, S_j^i, j = 1, …., n, of $\mathbf{S}^i$, represents a time series of data values collected over a period of time for the variables, $V_j, j = 1, \ldots, n$, respectively.
4. The domains of the n different variables, $V_j, j = 1, \ldots, n,$, are represented as domain $(V_j) = [L_{V_j}, U_{V_j}]$, $j = 1, \ldots, n$, respectively, so that L_{V_j} represent the lower bound and U_{V_j} represent the upper bound of the values that V_j can take on.
5. As $V_j, j \in \{1, \ldots, n\}$, is monitored over time, the values that V_j takes on at the time instants of $1, \ldots\ , p_{ij}$ can be represented as $S_j^i = (s_{j,1}^i,\ s_{j,2}^i, \ldots,\ s_{j,t-}\tau^i, \ldots,\ s_{j,t}^i, \ldots, s_{j,t+p_{ij}}^i)$, $1 \le \tau \le p_{ij}$, $\tau \in \mathbb{Z}^+$ and $s_{j,t}^i, t = 1, \ldots, p_{ij} \in \text{domain}(V_j)$.

Given a set of multivariate time series data, S, with characteristics as described above, the problem of clustering **S** is to find k clusters, C_1, $C_2, \ldots, C_k$, of **S**, where, $C_1 = \left\{\mathbf{S}_{C_1}^{(1)}, \mathbf{S}_{C_1}^{(2)}, \ldots, \mathbf{S}_{C_1}^{(n_1)}\right\}$, $C_2 = \left\{\mathbf{S}_{C_2}^{(1)}, \mathbf{S}_{C_2}^{(2)}, \ldots, \mathbf{S}_{C_2}^{(n_2)}\right\}$, …, $C_k = \left\{\mathbf{S}_{C_k}^{(1)}, \mathbf{S}_{C_k}^{(2)}, \ldots, \mathbf{S}_{C_k}^{(n_k)}\right\}$, where $\mathbf{S}_{C_j}^{(i)} \in \mathbf{S},\ i = 1, ldots, n_j$, and $j = 1, \ldots, k$, and $\bigcup_{j=1}^{k} C_j = \mathbf{S}$ and $C_j \cap C_{j'} = \emptyset$, and $j \neq j'$, and $j,\ j' \in \{1, \ldots, k\}$, in $\mathbf{S}$ so that the multivariate time series within a cluster is more similar to each other than those outside the cluster.

3.2 Intra- and Inter- CUVTS Temporal Patterns

The CUVTS in a MVTS are obtained by monitoring the values of a set of variables over a period of time. As these variables are temporally interrelated, the value that a particular variable take on at any time instant can be related to the variable's previous values or to the previous values of other variables. These interrelationships constitute, respectively, the intra-CUVTS and inter-CUVTS temporal patterns in a MVTS. These patterns can be compared against each other so that MVTS that exhibits similar patterns can be grouped together into the same cluster. To discover these clusters, one main task of MUTSCA is, therefore, to uncover the intra- or inter-CUVTS temporal patterns in each MVTS.

Given a value, say, $s_{j,t}^{i}$, in S_j^i within $\mathbf{S}^i$ that is observed at time, t, in order for MUTSCA to discover if it is temporally related to another value, $s_{j,t-\tau}^{i}$ previously observed at time, $t-\tau,\ 1 \leq \tau < \mathrm{t}$, MUTSCA determines how different the conditional probability, $\Pr(s_{j,t}^{i} | s_{j,t-\tau}^{i})$, is from the a priori probability $\Pr(s_{j,t}^{i})$. The larger the difference is, the more $s_{j,t}^{i}$ is temporally related to $s_{j,t-\tau}^{i}$. The differences in the probabilities between $s_{j,t}^{i}$ and previously observed values, $s_{j,t-\tau}^{i}$ ($\tau \leq \tau_{max}$, where τ_{max} is the maximum time-lag that a user chooses to explore) in S_j^i, therefore constitute the intra-CUVTS patterns of $\mathbf{S}^i$.

Similarly, given a value, say, $s_{j,t}^{i}$, in S_j^i within $\mathbf{S}^i$ that is observed at time, t, in order for MUTSCA to discover if it is temporally related to another value, $s_{j',t-\tau}^{i}$ in $S_{j'}^i$ previously observed at time, $t-\tau,\ 1 \leq \tau < \mathrm{t},\ S_{j'}^i \in \mathbf{S}^i$ and $S_j^i \neq S_{j'}^i$, MUTSCA determines how different the conditional probability, $\Pr(s_{j,t}^{i} | s_{j',t-\tau}^{i})$, is from the a priori probability, $\Pr(s_{j,t}^{i})$. The larger the difference is, the more $s_{j,t}^{i}$ is temporally related to $s_{j',t-\tau}^{i}$. The differences in the probabilities between $s_{j,t}^{i}$ and previously observed values, $s_{j',t-\tau}^{i}$, ($\tau \leq \tau_{max}$, where τ_{max} is the maximum time-lag that a user chooses to explore) in $S_{j'}^i$, $j' = 1, \ldots, n,\ j \neq j$, therefore constitute the inter- CUVTS patterns of $\mathbf{S}^i$.

For each MVTS, MUTSCA discover for it an associated set of intra-CUVTS and and inter-CUVTS temporal patterns. To decide if two MVTS should be grouped into the same cluster, we can compare their associated pattern sets to see if they are similar enough.

3.3 Transformation of Multivariate Time Series Data

To determine how large the differences are between the conditional probabilities, $\Pr(s^i_{j,t}|s^i_{j,t-\tau})$ and $\Pr(s^i_{j,t}|s^i_{j',t-\tau})$, and the a priori probability, $\Pr(s^i_{j,t})$, we will have to know the probability density functions and all parameters of them or to make assumptions about the data having a Guassian distribution, etc. In the absence of information about probability density functions or when the assumption of Guassian distribution cannot be validly made, we may have to estimate such probabilities directly from data. To reduce the number of parameters that need to be estimated, the approach we take to tackle the problem is to discretize the domain of all continuous variables being monitored into a relatively small finite number of intervals.

One commonly used technique for data discretization is to partition the domain of a continuous attribute into a finite number of intervals and assign a nominal value to each of them [19]. In other words, the domains of each the n different variables, $V_j, j = 1, \ldots, n$, $\text{domain}(V_j) = [L_{V_j},\ U_{V_j}]$, $j = 1, \ldots, n$, is to be partitioned into a relatively small finite number of intervals represented, respectively, as discretized $\text{domain}(V_j) = [I^1_{V_j}, I^2_{V_j}, \ldots, I^{n_j}_{V_j}], j = 1, \ldots, n$, respectively so that $\bigcup_{k=1}^{n_j} I^k_{V_j} = [L_{V_j}, U_{V_j}]$, and $I^k_{V_j} \cap I^{k'}_{V_j} = \emptyset$ if $k \neq k'$, $j = 1, \ldots, n$, respectively.

After discretization, the infinite number of possible continuous values that a variable being monitored can take on is now reduced into a small finite number of discrete values representing a small number of interval labels. The transformation of data significantly reduces the number of parameters that need to be estimated on one hand, it also no longer require relatively restrictive assumption on data and noise distributions on the other. Furthermore, the transforming of continuous into discrete data smooth out the data to reduce noise and hence improve accuracy, speeds up the clustering process and makes clustering results more meaningful and easier-to-understand [1].

Discretization algorithms can be classified into supervised and unsupervised [16]. In supervised discretization, we often assume that the class label is the ground truth but the class information could be questionable. On the other hand, although Equal Width [11] and Equal Frequency [11] does not require the class information, the selection of the number of intervals is not adequately addressed and their criteria of discretization fail to consider the relationship between the interval boundaries and the correlated attributes if they exist. Hence, in this paper, we use the discretization algorithm provided in [16] which is an unsupervised discretization algorithm but use an information measure that reflects interdependence between the continuous attribute and the representative attribute in an attribute groups. To minimize the effect of noise in the clustering process, rather than the actual continuous values, the data is partitioned into intervals (levels) [13]. The partitioning, which is also called discretization, is based on a popular technique as described in [16] so as to minimize the loss of information during the process.

3.4 Discovering CUVTS Temporal Patterns

Lift Ratio are supposed to interact with each other over time for each CUVTS. The discovering of the patterns of interaction reflects the interaction of the time series that are being monitored. To discover the interrelating patterns among the time series, it tries to detect for temporal association relationship (TAR) among the data values between CUVTS.

For CUVTS, we will discover both intra- and inter- CUVTS temporal patterns, the intra- patterns consider the relationship between the same CUVTS but different time points. And the *Lift Ratio* can be estimated for intra- patterns as follows:

$$\mathit{Lift}\ (s^i_{j,t-\tau} = > s^i_{j,t}) = \frac{\Pr\left(s^i_{j,t}|s^i_{j,t-\tau}\right)}{P\left(s^i_{j,t}\right)} = \frac{\mathit{freq}(s^i_{j,t-\tau} \wedge s^i_{j,t})}{\mathit{freq}\left(s^i_{j,t}\right) * \mathit{freq}\left(s^i_{j,t-\tau}\right)}(1 \le \tau < \mathrm{t})$$

And the inter-CUVTS temporal patterns can also be determined in very much the same way. Given a value, say, $s^i_{j,t}$, in S^i_j in $\mathbf{S}^i$, and another value, say, $s^i_{j',t-\tau}$, in $S^i_{j'}$ in $\mathbf{S}^i$, The algorithm discovers the *Lift Ratio* as *Lift* $(s^i_{j',t-\tau} = > s^i_{j,t})(1 \le \tau < \mathrm{t})$.

For example, after calculating lift ratio, we can discover the significant rules like if $s^i_{j',t-\tau}$ is always preceded at $\tau(\tau \ge 0)$ positions earlier by $s^i_{j,t}$, one can conclude that $s^i_{j,t}$ is depended on $s^i_{j,t}$. If so, the proposed algorithm considers $s^i_{j,t}$ as temporally related to $s^i_{j',t-\tau}$. The set of all values, $s^i_{j',t-\tau}$, $1 \le \tau < \mathrm{t}$, that $s^i_{j,t}$ is temporally related to, constitutes the inter-CUVTS patterns of $\mathbf{S}^i$.

And the pseudo code for clustering multivariate temporal clustering algorithm is given in Fig. 1.

```
Proposed Algorithm:
Input : S= $\{\mathbf{S}^1, \mathbf{S}^2, \ldots, \mathbf{S}^N\}$.,variables $\mathbf{S}^i = \{S^i_1, S^i_2, \ldots, S^i_n\}$.
Output : a set of ì Lift Ratioî  values for each MVTS
For each multivariate time series
    Discretized data using Equal Frequency for $\mathbf{S}^i$
    For each variables ($v_1$) in MVTS
        get intra-patterns $Lift\ (s^i_{j,t-\tau} => s^i_{j,t})$ ($1 \le \tau <$ t)
        For each variables $v_2$ besides$v_1$ in MVTS
            get inter-patterns $Lift\ (s^i_{j',t-\tau} => s^i_{j,t})$ ($1 \le \tau <$ t)
            Result + = inter-patterns
            (Result is one set of Lift Ratio for one MVTS)
        End
        Result + = inter-patterns
    End
  finalResult += Result
End  (finalResult are allLift Ratio for all MVTSs)

K-means clustering continue.
```

Fig. 1. The pseudo code for pattern discovery from MVTS data

```
Input: k (the number of clusters),
       D (a set of lift ratios)
Output: a set of k clusters
Method:
Arbitrarily choose k objects from D as the initial cluster centers;
Repeat:
  1. (re)assign each object to the cluster to which the object is
  the most similar, based on the mean value of the objects in the
  cluster;
  2. Update the cluster means, i.e., calculate the mean value of
  the objects for each cluster
Until no change;
```

Fig. 2. The pseudo code for K-means clustering algorithm

After transferring original MVTSs into a set of lift ratio, K-means clustering algorithm [1] is used for clustering the set of lift ratio and each MVTS can be treated as an object. In this case, the value of *lift ratio* can determine how relationship between the values of different variables for one MVTS at the same or different time points, hence we use a set of value of lift ratio to describe original MVTS data and treat each MVTS as a record using the value of lift ratio, and cluster the transformation data by using K-means clustering algorithm.

First, it randomly selects k of the objects, each of which initially represents a cluster mean or center. For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object. It then computes the new mean for each cluster. This process iterates until the clusters has no changes. The algorithm of k-mean is like Fig. 2. Pseudo code shows.

4 Performance Evaluation

To evaluate the performance of proposed algorithm, two different datasets experiments were carried out using both synthetic and real world data. The set of synthetic data that was used was generated by embedding different patterns in the data to see if the proposed algorithm can discover these patterns for clustering. The real-world dataset is a set of EMG data. For the purpose of performance evaluation, the known clustering arrangements were compared with those found by proposed algorithm using two performance measures, the *clustering accuracy* and the *F*-measure. In the following, we describe the characteristics of datasets and explain how they were used in our experiments for performance analysis of proposed algorithm. We also present and discuss the experimental results in details.

4.1 Experiments Using Synthetic Data

The set of synthetic data used to evaluate the performance of proposed algorithm consists of 45 multivariate time series that were generated to belong to three clusters,

C_1, C_2 and C_3 and each of clusters contains 15 multivariate time series (MVTS), making up a total of 45 MVTS. Each of these multivariate time series consist, in turn, of 4 component univariate time series (CUVTS). For each MVTS in each cluster, both inter- and intra-CUVTS patterns were generated and embedded in such a way that the MVTS that are in the same cluster have more similar patterns than those that are in the other clusters.

To describe the patterns embedded in the time series data, let us assume that the values of the four CUVTS of each MVTS are collected as a result of monitoring the values of four different variables, $v_1, v_{2,} v_{3,}$ and v_4, respectively, at successive instants spaced at uniform time intervals over a period of time. The values of these variables are assumed to be normalized so that they are all within the interval from 0 to 1, i.e., $domain(v_i) = [0, 1]$, i = 1, …, 5. For our experiments, a total of 200 data values are generated for each of the variables and hence each of the 5 CUVTS consists of 200 data points.

For the MVTS in C_1, the values of the four CUVTS are generated in the following ways: (i) the values corresponding to v_1 are generated randomly, (ii) the values corresponding to v_4 are generated in such a way that at every interval of 5 (determined stochastically) time instants, v_4 takes on a value in the interval [0.18, 0.44] at the following time instant and whenever it does so, v_2 takes on is generated to be within [0.03, 0.39], (iii) if the values of v_4 is not in the interval [0.18, 0.44], v_3 takes on values in [0.32, 0.49] at the next time. Similarly, for the MVTS in C_2, the values of the four CUVTS are generated in the following ways: (i) v_2 are generated randomly, (ii) if v_2 is in [0.03, 0.50] then v_1 is generated to be within [0.53, 0.86], (iii) if v_2 is [0.40, 0.66], v_4 will take on a value within [0.23, 0.41] at the next time points 60 % of the time, and a value within [0.05, 0.49] 40 % of the time. For the MVTS in C_3: (i) v_3 are generated randomly, (ii) if v_3 is in [0.03, 0.22] then v_2 is generated to be within [0.53, 0.92], (iii) if v_1 is in [0.70, 0.90] at every six time instants, v_4 will in [0.13, 0.47] at the next time point.

For this synthetic data set, as the cluster membership for each MVTS is already known, we can use such performance measures clustering accuracy and the F-measure [7] to evaluate the performance of algorithm for the datasets which knowing class labels. As the correct cluster membership of each MVTS is already known, the F-measure would provide objective information on the degree to which the algorithm can correctly cluster the data [7]. The F-measure is defined as $F\left(C_p, C_q\right) = \frac{2Recall(Cp,Cq)Predcision(Cp,Cq)}{Recall(Cp,Cq)+Precision(Cp,Cq)}$, where $Recall(C_p, C_q) = \frac{count_{Cp,Cq}}{count_{Cp}}$, $Predcision(Cp, Cq) = \frac{count_{Cp,Cq}}{count_{Cq}}$ and $count_{Cp,Cq}$ is the number of records with cluster label Cp in the discovered cluster Cq, $count_{Cp}$ is the number of records with cluster label Cp and $count_{Cq}$ is the number of records in the discovered cluster Cq. The F-measure has value in the interval [0,1] and the large its value, the better the clustering quality it reflects. Finally, we will calculate the accuracy of all F-measures for each cluster and use it to evaluate the performance of MUTSCA. Similarly, the clustering accuracy can be defined as following:

$$Clustering\ Accuracy = \frac{\sum_{i=1}^{n} count_{ci}}{Total\ samples\ number},\ i = 1, \ldots n\ (n\ is\ the\ number\ of\ clusters)$$

For performance benchmarking, we attempted to compare proposed algorithm with other model-based algorithms corresponding to the use of ARMA and HMM model. As existing algorithms are mainly developed to tackle univariate time series clustering, we will use PCA to extract features from each MVTS and transform them into a single univariate time series data. And then model the univariate time series data using ARMA and the first eight LPC coefficients to represent each univariate time series data. Once we have the LPC coefficients determined, we use the *k*-means with the *Euclidean* distance to cluster the data. An alternative approach to the model-based approach has also been implemented.

Table 1. Comparison for different algorithms by using synthetic dataset

Approaches	F-measure	Clustering accuracy
Model based (PCA + ARMA)	0.503	51.1 %
Model based (PCA + HMM)	0.428	46.67 %
MUTSCA (LR)	0.8	86.67 %

Instead of the ARMA, we used another model called HMM model. We discretize continuous data into 3 intervals after PCA transformation and then the 45 time series data will be initialized into three clusters. For each cluster, a HMM model is created and the value of *log-likelihood* will be calculated for each time series and the three initial models. The lower value of log-likelihood, the more closed between time series and model. After 100 iterations, the final cluster will be determined. The results of our performance evaluation experiments using MUTSCA are given in Table 1. We use the *lift ratio* to represent each MVTS which is called as Multivariate time series clustering (LR). As shown in the Table, for MUTSCA, we have a clustering accuracy of 86.7 % and an average F-measure of 0.792 for the proposed algorithm.

The other two approaches which are treated as model-based approaches and ARMA and HMM models are used. In PCA + ARMA approach, the accuracy is also low, say 51.1 % and the average of F-measure is just 0.503. The reason why the accuracy of method PCA + ARMA is also lower one because there are 8 coefficients of ARMA is used to describe the original data. Lots of information will be lost when high-dimension is reduced. Then the other model-based algorithm – HMM is used as another comparison. The accuracy is just 46.67 % with the average of F-measure is 0.428. However, all of above methods just deal with original data in the view of statistic but not consider the relationship between different variables. After avoiding this disadvantage, our proposed method can deal with this kind of data effectively, and all the data can be clustered into correct cluster. Therefore, the accuracy and F-measure are higher for proposed algorithm.

4.2 Experiments Using Vicon Physical Actions Data Set

The other real-world dataset [6] consists of data collected from seven male and three female subjects aged between 25 to 30 who had experienced aggression in such scenarios as physical fighting, took part in the experiments. A total of 20 individual experiments were carried out with each subject performing ten normal and ten aggressive activities. The normal activities were bowing, clapping, handshaking, hugging, and jumping while the aggressive activities were elbowing, front kicking, hammering, headering, and kneeing.

The subjects' performance has been recorded by the Vicons nine ubiquitous cameras, interfacing human activity with spatial coordinate points. Based on this context, the data acquisition process involved four reflectable markers placed on the forearms (elbows and wrists), four on the forelegs (knees and ankles), and one on the top of the head. A pair of markers (except the head) is attached at each body segment for 3D data acquisition. - Arm markers: wrist (WRS), elbow (ELB) - Leg markers: ankle (ANK), knee (KNE) Coords: The 3 coordinates (x,y,z) define the 3D position of each marker in space. Each file in the dataset contains in overall 28 columns (the 1st is a counter), and is organised as follows: Segment: A segment defines a body segment or limb. From the overall number of markers, there are 27 input time series for all the three x, y, and z coordinates (9 markers × 3 coordinates). Each marker-coordinate time series contains $\sim$3,000 samples. In summary, this experiment data is a 100 MVTSs data (10 people × 10 actions) × 27 variables × $\sim$ 3000 samples multivariate time series database. In our study, we cluster all these activities experiment into two clusters, "Normal" and "Aggressive", and we use original class label "Normal" and "Aggressive" which is like the work of paper [3] has been done to test clustering accuracy for each experiment. We use 7 males and 3 females as our experiment subject, and all of them need finish 10 actions which include 5 normal actions and 5 aggressive actions for each experiment and there are 27 input variables for each experiment. We clustering all these multivariate time series data into two clusters, "Normal Actions" and "Aggressive Actions", so K = 2 for K-means clustering. The clustering accuracy result has been given in Table 2.

Our proposed algorithm is also getting the highest accuracy of 81 % (F-measure 0.8) for proposed algorithm. However, if we consider PCA as extraction tool and ARMA model for the data, the accuracy is reduced to around 50 % and the accuracy is 72 % using HMM model. And the F-measure values are 0.41 and 0.72 respectively. This experiment result can also prove that the proposed algorithm can get the highest accuracy and F-measure.

Table 2. Comparison for different Algorithms by using EMG dataset

Approaches	F-measure	Clustering accuracy
Model based(PCA + ARMA)	0.41	51.1 %
Model based (PCA + HMM)	0.72	72 %
MUTSCA (LR)	0.8	81 %

5 Conclusion

In the view of data mining algorithm, the contributions of the proposed algorithm can be characterized as below. Firstly, our proposed clustering algorithm can deal with discretized data or even mixed data if we discretize the continuous data in original dataset. According to discretization, the noisy data can be avoided. Secondly, we consider the relationship between different variables value by using “Lift Ratio” that means we didn’t consider only one variable each time or just set a weight for different variables but consider the significant value for different variables. The above two points reveals we can not only avoid noisy data but also can save useful information as more as possible. Besides these, the third advantage of proposed methods is that we can continue do clustering using K-means, or hierarchical clustering algorithms after transforming original data into a set of lift ratio value. In summary, our proposed method can deal with a set of unequal length, multivariate, discretized value time series data.

We will treat implement clustering algorithm with fuzzy discretized methods as our future work, which can improve clustering accuracy. In this research, we pay more attention to the description of multivariate time series data, and just use K-means clustering algorithm which is the simplest one as our clustering algorithm. The experiment result shows, the classification accuracy is so much higher than just only using K-means clustering algorithm. Besides, we will also consider data stream motif as well as graph stream mining continues. We have the confidence to find out better algorithm to improve our proposed method.

References

1. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Elsevier Inc, Amsterdam (2007)
2. Liao, T.W.: Clustering of time series data - a survey. Pattern Recogn. **38**, 1857–1874 (2005)
3. Theodoridis, T., Hu, H.: Classifying aggressive actions of 3D human models using dynamic ANNs for mobile robot surveillance. In: IEEE International Conference on Robotics and Biomimetics (Robio-2007), pp. 371–376, 15–18 December 2007
4. Wang, X.Z., Smith, K., Hyndman, R.: Characteristic-based clustering for time series data. J. Data Min. Knowl. Discov. **13**(3), 335–364 (2006)
5. Verdoolaege, G., Rosseel, Y.: Activation detection in event-related fmri through clustering of wavelet distributions. In: IEEE 17th International Conference on Image Processing, Hong Kong, pp 4393–4395 (2010)
6. Asuncion, A., Newman, D.J.: UCI machine learning repository. Irvine, CA: University of California, School of Information and Computer Science (2007). http://archive.ics.uci.edu/ml/
7. Larsen, B., Aone, C.: Fast and effective text mining using linear-time document clustering. In: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 16–22 (1999)
8. Rani, S., Sikka, G.: Recent techniques of clustering of time series data: a survey. Int. J. Comput. Appl. **52**(15), 1–9 (2012)
9. Li, C., Biswas, G., Dale, M., Dale, P.: Building models of ecological dynamics using HMM based temporal data clustering - a preliminary study. In: Hoffmann, F., Adams, N., Fisher, D., Guimaraes, G., Hand, D.J. (eds.) IDA 2001. LNCS, vol. 2189, pp. 53–62. Springer, Heidelberg (2001)
10. Wang, L., Mehrabi, M.G., Kannatey-Asibu Jr., E.: Hidden Markov model-based wear monitoring in turning. J. Manufact. Sci. Eng. **124**, 651–658 (2002)

11. Wong, A.K.C., Wang, C.C.: DECA – a discrete-valued ensemble clustering algorithm. IEEE Trans. Pattern Anal. Mach. Intell. **1**, 342–349 (1979)
12. Guo, C., Jia, H., Zhang, N.: Time series clustering based on ICA for stock data analysis. Wireless Communications, Networking and Mobile Computing, WiCOM '08, pp. 1–4 (2008)
13. Ma, P.C.H., Chan, K.C.C., Chiu, D.K.Y.: Clustering and re-clustering for pattern discovery in gene expression data. J. Bioinform. Comput. Biol. **3**(2), 281–301 (2005)
14. Ye, J., Janardan, R., Li, Q.: Gpca: an efficient dimension reduction scheme for image compression and retrieval. In: KDD '04: The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, pp. 354–363 (2004)
15. Kalpakis, K., Gada, D., Puttagunta, V.: Distance measures for effective clustering of ARIMA time-series. In: IEEE International Conference on Data Mining, San Jose, CA, pp. 273–280 (2001)
16. Wong, A.K.C., Wu, B., Wu, G.P.K., Chan, K.C.C.: Pattern discovery for large mixed-mode database. In: CIKM'10, pp. 859–868, 26–30 October 2010
17. Xiong, Y., Yeung, D.-Y.: Mixtures of ARMA models for model-based time series clustering. In: Proceedings of the IEEE International Conference on Data Mining, Maebaghi City, Japan, 9–12 December 2002
18. Shumway, R.H.: Time–frequency clustering and discriminant analysis. Stat. Probab. Lett. **63**, 307–314 (2003)
19. Dimitrova, E.S., McGee, J.J., Laubenbacher, R.C.: Discretization of Time Course Data (2005). http://polymath.vbi.vt.edu/discretization/DimitrovaMcGeeLaubenbacher.pdf

Mining Time-Aware Transit Patterns for Route Recommendation in Big Check-in Data

Hsun-Ping Hsieh(✉) and Cheng-Te Li

Graduate Institute of Networking and Multimedia,
National Taiwan University, Taipei 106, Taiwan
{d98944006,d98944005}@csie.ntu.edu.tw

Abstract. In current location-based services, there are numerous travel route patterns hidden in the user check-in behaviors over locations in a city. Such records rapidly accumulate and update over time, so that an efficient and scalable algorithm is demanded to mine the useful travel patterns from the big check-in data. However, discovering travel patterns under efficiency and scalability concerns from large-scaled location data had not ever carefully tackled yet. In this paper, we propose to mine the *Time-aware Transit Patterns* (TTP), which capture the representative traveling behaviors over consecutive locations, from the big check-in data. We model the travel behaviors among different locations into a *Route Transit Graph* (RTG), in which nodes represents locations, and edges denotes the transit behaviors of users between locations with certain time intervals. The *time-aware transit patterns*, which are required to satisfy frequent, closed, and connected requirements due to respectively physical meanings, are mined based on the RTG transaction database. To achieve such goal, we propose a novel TTPM-algorithm, which is devised to only need to scan the database once and generate no unnecessary candidates, and thus guarantee better time efficiency lower and memory usage. Experiments conducted on different cities demonstrate the promising performance of our TTPM-algorithm, comparing to a modified *Apriori* method.

Keywords: Time-aware transit patterns · Check-in data · Route planning

1 Introduction

Nowadays, location-based services (LBS), such as Foursquare[1] and Gowalla,[2] keep track of personal geospatial journeys through check-in actions. With smart phones, users can easily perform check-in actions, and the geographical information of locations with timestamps is stored in LBS. Eventually a large-scaled user-generated location sequences (i.e., routes) data are derived. Such location sequence data can not only collectively represent the real-world human geo-activities, but also serve as a handy resource for constructing location-based recommendation systems. Since the user-moving records implicitly reveal how people travel around an area with rich

[1] https://foursquare.com/

[2] http://gowalla.com/

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 818–830, 2014.
DOI: 10.1007/978-3-319-13186-3_73

spatial and temporal information, including longitude, latitude, and recording timestamp, one reasonable application leveraging such user-generated location sequence data is to recommend travel routes. While existing trip planning systems [6, 17] consider either the shortest geodesic distance or the shortest time period to plan routes, we believe it would be more useful if the *representative* routes with visiting time information can be recommended from the user check-in data.

Although LSBs possesses rich user check-in behavioral data, they suffer from two major problems such that it is hard to directly use the routes of users for recommendation. First, the check-in behaviors of local residents and the travelers are different. How to find the significant traveling routes is a critical issue. Second, the check-in data contains diverse kinds of noises. For example, some outlier users perform check-in frequently (up to one hundred times) within a day. And there could have a number of missing locations during the trips of users. In this paper, to provide a better travel recommendation, we aim to devise an unsupervised mechanism that automatically summarizes the representative travel patterns. We propose to mine a novel pattern, *Time-aware Transit Patterns* (TTP), which are the representative time-labeled routes mined from the check-in data. A TTP is a sequence of locations, in which each transit between locations is associated with a visiting timestamp. For example, an example TTP is <(MMOA museum, Macy's store, 0, 3), (Macy's store, H&M store, 3, 5), (H&M store, Time Square, 5, 6)>, which refers to that the recommended route consists of MMOA museum, Macy's store, H&M store and Time Square in order. And the pattern suggests that the staying time as well as the transportation time between MMOA museum and Macy's store is $3 - 0 = 3$ h, and so on. Note that the TTP focuses on the *relative* timestamp which is more flexible than absolute timestamp. We believe finding such notable travel patterns can not only allow us perform route planning in an effective manner but also benefit transportation scheduling.

We use the graph structure to represent the transit behaviors between locations. We propose *Route Transit Graph* (RTG) to model the location transitions that are generated by user-generated routes. RTG is a directed graph, in which each node represents for a location and each edge stands for a transit from one location to the other. Besides, each edge is associated with a set of time intervals that record the durations of users' location transitions. Each time interval consists of a starting timestamp and an arrival timestamp. On the other hand, considering that the real-world travel activities usually follow a certain reasonable periodic property, such as that each time users usually plan one-day routes when traveling, we construct a RTG for the check-in records within a day. By collecting RTGs corresponding to days in the entire check-in data, we can construct a RTG transaction database.

In this paper, we aim to mine the *Time-aware Transit Patterns* (TTP) from the constructed RTG transaction database. Each mined TTP can be regarded as a representative route for the recommendation. Based on PrefixSpan [19] and BIDE [13], we propose an efficient and scalable algorithm, *Time-aware Transit Pattern Mining algorithm* (TTPM-algorithm), to mine the closed frequent Time-aware Transit Patterns from the constructed RTG transaction database. In addition, in the proposed TTPM-algorithm, we exploit the *closed pattern* mining concept to reduce the number of frequent patterns during mining process so that the time efficiency can be boosted and the memory usage can be well utilized.

2 Related Work

When considering temporal pattern mining methods, sequential pattern mining is one of the traditional mining methods to mine the frequently-occurring temporal events or subsequences. The sequential pattern mining problem was first introduced by Agrawal et al. [2]. They proposed a method, called GSP, which adopts a generate-and-test approach based on the Apriori method to mine frequent sequential patterns. Since then, many sequential pattern mining methods have been proposed, e.g., SPAM [3], SPADE [18], GO-SPADE [8], and PrefixSpan [10]. SPAM [3] exploits a vertical bitmap structure to count supports efficiently. SPADE [18] and GO-SPADE [8] uses a vertical data format and a divide-and-conquer strategy to reduce the search space and the number of database scans. PrefixSpan [10] uses the projected database to mine the complete set of patterns and to reduce the efforts of candidate subsequence generation. On the other hand, many real-world applications can be modeled as graphs such as chemical structure, web structures, social networks, etc. Therefore, mining frequent subgraphs has become an important issue in the data mining area. AGM [5] and FSG [7] use a level-wise approach to mine frequent subgraphs, which combines the frequent subgraphs mined at the previous level to generate all candidates at the next level. gSpan [14] discovers all frequent patterns without candidates generation and forms a canonical labeling system to mine frequent subgraphs.

Since mining all frequent patterns may consume lots of time and storage, Pasquier et al. [9] introduced a new concept to mine the frequent closed patterns. A frequent pattern is closed if there does not exist any super-pattern with the same support. Yan et al. [15] proposed CloSpan [15] method to mine closed sequential patterns by candidate maintenance-and-test paradigm to prune the search space and check if a newly found frequent sequence is promising to be closed. Wang et al. [13] presented an algorithm called BIDE to improve the performance of CloSpan [15] without keeping track of any frequent closed patterns (or candidates) for closure checking. Yan et al. [16] designed an approach, called CloseGraph [16], to find frequent closed graph patterns which can reduce unnecessary subgraphs to be generated.

3 Problem Definition

We represent the transition behaviors of users between check-in locations using the proposed Route Transit Graph (RTG). A RTG is a directed labeled graph, which is defined as $g = (V, E, \Sigma, \lambda)$, where V is a set of vertices, E is a set of directed edges, $(i, j) \in E$ is an edge from node i to node j, Σ is a set of labels, λ is a label function, λ: $V \cup E \rightarrow \Sigma$, and the label function assigns a label to a vertex or an edge. Each edge is associated with a list of timestamps. Thus, an edge can be presented by a 4-tuple, (S, D, t_s, t_e), where S and D are the vertices, and t_s is the starting time of the source location, and t_e is the arrival time of the destination. For example, the transit edge from vertex L_1 to vertex L_2 during timestamp [11, 13] (unit: hour) is represented by $(L_1, L_2, 14, 16)$. Considering a graph database DB containing n RTG graphs, $DB = \{g_1, g_2, \ldots, g_n\}$, where g_i is a directed labeled graph, $1 \leq i \leq n$.

We sort the edges by the associated timestamps and transform the graph into an edge sequence. Figure 1 illustrates an example RTG graph database. For example, the graph g_1 can be represented as an edge sequence <(L_4, L_3, 8, 10), (L_3, L_5, 13, 17), (L_1, L_2, 14, 16), (L_2, L_3, 16, 17), (L_2, L_4, 16, 17), (L_2, L_6, 17, 18), (L_4, L_5, 18, 19), (L_5, L_6, 20, 23)>. The edges sequence are sorted the order mention by Definition 1.

Definition 1 (Edge Order). Let $\alpha = (S_1, D_1, t_{s1}, t_{e1})$ and $\beta = (S_2, D_2, t_{s2}, t_{e2})$ be two edges in a graph. $\alpha < \beta$ if (1) $t_{s1} < t_{s2}$, (2) $t_{s1} = t_{s2}$ and $t_{e1} < t_{e2}$, (3) $S_1 < S_2$, $t_{s1} = t_{s2}$, and $t_{e1} = t_{e2}$, or (4) $D_1 < D_2$, $S_1 = S_2$, $t_{s1} = t_{s2}$, and $t_{e1} = t_{e2}$.

Definition 2 (Edge Containment). A transit edge $\alpha = (S_1, D_1, t_{s1}, t_{e1})$ is contained by another edge $\beta = (S_1, D_1, t_{s2}, t_{e2})$ if $t_{s2} \le t_{s1}$ and $t_{e1} \le t_{e2}$, denoted as $\alpha \subseteq \beta$. For example, if $\alpha = (L_1, L_2, 1, 2)$, $\beta = (L_1, L_2, 0, 3)$, α is contained by β.

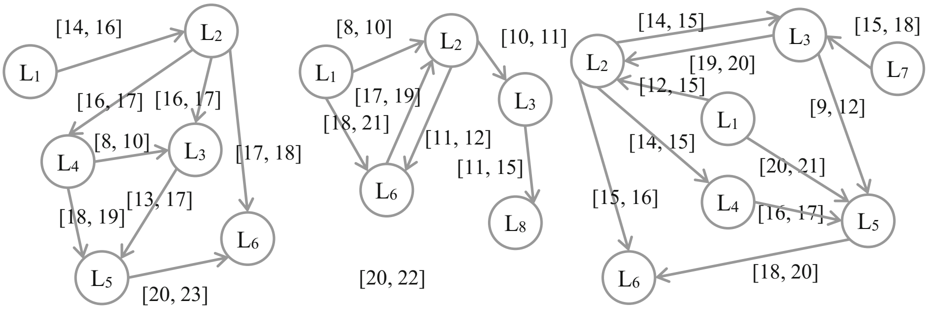
Definition 3 (Time-aware Transit Pattern). A Time-aware Transit Pattern is defined as <(S_1, D_1, t_{s1}, t_{e1}), (S_2, D_2, t_{s2}, t_{e2}), …, (S_h, D_h, t_{sh}, t_{eh})>, where $t_{s1} = 0$, and all the edges in the pattern are sorted in increasing order. The time-aware transit pattern should satisfy *route connected property* which is defined in Definition 4.

Definition 4 (Route Connected Property). A route transit sequence <(S_1, D_1, t_{s1}, t_{e1}), (S_2, D_2, t_{s2}, t_{e2}), …, (S_h, D_h, t_{sh}, t_{eh})> follows route connected property if $\forall\ S_i \ni S_2 \cup S_3 \cup \ldots \cup S_h$ we can always find an edge ej whose $j < i$, and $D_j = S_i$ or $S_1 = S_i$.

Definition 5 (Pattern Length). The length of a pattern is defined as the number of edges. A pattern of length k is called a k-pattern.

Definition 6 (Pattern Existence). A pattern <(ps_1, pd_1, pt_{s1}, pt_{e1}), (ps_2, pd_2, pt_{s2}, pt_{e2}), …, (ps_m, pd_m, pt_{sm}, pt_{em})> is contained by a graph <(gs_1, gd_1, gt_{s1}, gt_{e1}), (gs_2, gd_2, gt_{s2}, gt_{e2}), …, (gs_n, gd_n, gt_{sn}, gt_{en})> if there exists a sequence of integers $j_1 < j_2 < \ldots < j_n$ so that $pu_i = gu_{ji}$, $pl_i = gl_{ji}$, $pv_i = gv_{ji}$, $pt_{si} \ge gt_{sji} - gt_{sj1}$, and $pt_{ei} \le gt_{eji} - gt_{sj1}$, $i = 1, 2, \ldots, n$.

For example, P = <(L_4, L_5, 0, 1), (L_5, L_6, 2, 5)> is contained by the graph g_1 shown in Fig. 1, where $j_1 = 7$, $j_2 = 8$.



(a) g_1. (b) g_2. (c) g_3.

Fig. 1. An example of RTG database.

Since a pattern may contain many edges, discovering all such patterns would be lack of efficiency and not useful for route planning applications. Therefore, we define some constraints to reduce unnecessary ones. For example, P = <(L_4, L_3, 0, 2), (L_3, L_6, 9, 10)> is not helpful since the time span between two edges are too large. The user may only be interested in the patterns whose time spans are within a certain time interval. Therefore, we define a maximum time-span threshold *maxgap*, in Definition 7.

Definition 7 (Timespan Threshold: *maxgap*). The timespan between two edges (S_i, D_i, t_{si}, t_{ei}) and (S_k, D_k, t_{sk}, t_{ek}) is defined as $|t_{sk} - t_{ei}|$. We aim to mine the patterns which follow the *maxgap* constraint: for each k^{th} edge e_k ($k \geq 2$) in the pattern P, we must find another edge e_i from e_1 to e_{k-1} that make $|t_{sk} - t_{ei}| \leq maxgap$, where *maxgap* is a user-specified threshold. We should notice that t_{ei} is sometimes larger than t_{sj} since we allow flexible route construction. For example, P = <(L_4, L_3, 0, 4), (L_3, L_6, 3, 5)> suggests that users can arrive location L_3 around 3–4 h.

Definition 8 (Super-pattern). A pattern <(ps_1, pd_1, pt_{s1}, pt_{e1}), (ps_2, pd_2, pt_{s2}, pt_{e2}), …, (ps_m, pd_m, pt_{sm}, pt_{em})> is a super-pattern of another pattern <(qs_1, qd_1, qt_{s1}, qt_{e1}), (qs_2, qd_2, qt_{s2}, qt_{e2}), …, (qs_n, qd_n, qt_{sn}, qt_{en})> if there is a sequence of integers $j_1 < j_2 < \ldots < j_n$ so that $ps_i = qs_j$, $pd_i = qd_{ji}$, $qt_{si} \geq pt_{sji} - pt_{sj1}$, and $qt_{ei} \leq pt_{eji} - pt_{sj1}$, $i = 1, 2, \ldots, n$.

For example, P = <(L_1, L_2, 0, 2), (L_2, L_4, 2, 3), (L_4, L_5, 4, 5), (L_5, L_6, 6, 9)> is a super-pattern of Q = <(L_1, L_2, 0, 2,) (L_2, L_4, 2, 3), (L_4, L_5, 4, 5)>, where $j_1 = 1$, $j_2 = 2$ and $j_2 = 3$. Moreover, P = <(L_1, L_2, 0, 2), (L_2, L_4, 2, 3), (L_4, L_5, 4, 5), (L_5, L_6, 6, 9)> is a super-pattern of Q = <(L_1, L_2, 0, 2), (L_2, L_4, 2, 3), (L_4, L_5, 4, 5), (L_5, L_6, 6, 7)>, where $j_1 = 1$, $j_2 = 2$, $j_3 = 3$ and $j_4 = 4$.

Definition 9 (Pattern Support). The *support* of a pattern P, denoted as $sup(P)$, is defined as the number of graphs containing P in the database.

Definition 10 (Frequent Pattern). A pattern P is *frequent* if $sup(P)$ is not less than *minsup*, where *minsup* is a user-specified minimum support threshold.

Definition 11 (Closed pattern). A frequent pattern P is *closed* if there does not exist any super-pattern of P with the same support.

For example, in Definition 8, if P = <(L_1, L_2, 0, 2) (L_2, L_4, 2, 3) (L_4, L_5, 4, 5) (L_5, L_6, 6, 9)> is a super-pattern of Q = <(L_1, L_2, 0, 2) (L_2, L_4, 2, 3) (L_4, L_5, 4, 5)> and both their support are the same. Thus, Q is not closed.

Definition 12 (Prefix & Postfix). Given a pattern P = <(S_1, D_1, t_{s1}, t_{e1}), (S_2, D_2, t_{s2}, t_{e2}), …, (S_m, D_m, t_{sm}, t_{em})>, Q = <(S_1, D_1, t_{s1}, t_{e1}), (S_2, D_2, t_{s2}, t_{e2}), …, (S_i, D_i, t_{si}, t_{ei})> is called a prefix of P, and R = <(S_{i+1}, D_{i+1}, t_{si+1}, t_{ei+1}), (S_{i+2}, D_{i+2}, t_{si+2}, t_{ei+2}), …, (S_m, D_m, t_{sm}, t_{em})> is called the postfix of P, $1 \leq i \leq m$.

Definition 13 (Projected Database). The P-projected database, denoted as $DB|_p$, contains all postfixes of the graph possessing P in database DB, where P is a frequent pattern in DB.

Assume a pattern P = <(L_1, L_2, 0, 2)>. In Fig. 1, since <(L_1, L_2, 0, 3)> contains <(L_1, L_2, 0, 2)>, and <(L_1, L_2, 1, 3)> and these two edges can be considered as the pattern <(L_1, L_2, 0, 2)> by shifting the time interval to [0, 2], there are two postfixes of

P in g_3. For the prefix <(L_1, L_2, 0, 2)>, the postfix is <(L_2, L_3, 2, 3), (L_2, L_4, 2, 3), (L_2, L_6, 3, 4) (L_7, L_3, 3, 6), (L_4, L_5, 4, 5), (L_5, L_6, 6, 8), (L_3, L_2, 7, 8), (L_1, L_5, 8, 10)>. For the prefix <(L_1, L_2, 1, 3)>, the corresponding postfix is <01(L_2, L_3, 1, 2), (L_2, L_4, 1, 2), (L_2, L_6, 2, 3), (L_7, L_3, 2, 5), (L_4, L_5, 3, 4), (L_5, L_6, 5, 7), (L_3, L_2, 6, 7), (L_1, L_5, 7, 9)>. Similarly, there is one postfix of P in g_1, namely <(L_2, L_3, 2, 3), (L_2, L_4, 2, 3), (L_2, L_6, 3, 4), (L_4, L_5, 4, 5) (L_5, L_6, 6, 9)>.

Definition 14 (Concatenation Function). The concatenation of patterns P and Q is denoted as $P \oplus Q$. For example, if P = <(L_1, L_2, 0, 2)> and Q = <(L_2, L_3, 2, 3)>, $P \oplus Q$ = <(L_1, L_2, 0, 2), (L_2, L_3, 2, 3)>.

4 Methodology

The proposed algorithm consists of two stages. First, we mine all frequent patterns of length one (denotes 1-patterns), P in the database. Next, for each frequent k-pattern ($k \geq 1$) found P, we build the projected database for each frequent k-pattern found. And then we scan its projected database to find local frequent 1-patterns e which is connected with P. For each local frequent 1-pattern e, we concatenate P with e to form a frequent $(k + 1)$-pattern. The concatenations are recursively performed in a depth-first search manner until no more frequent closed patterns can be found. During the mining process, we use the closure checking and pruning strategies to reduce impossible and unnecessary candidates. Thus, the proposed algorithm can efficiently mine closed frequent Time-aware Transit Patterns in a RTG database.

4.1 Frequent Patterns Enumeration

We introduce a TTP-tree to enumerate frequent patterns where each node represents a frequent pattern, and the level at which the node is located represents the length of the frequent pattern. For example, the frequent 1-patterns are recorded at level 1 and the frequent 2-patterns are at level 2. Moreover, a pattern at level k is derived from the pattern of its parent node at level $k - 1$, $k \geq 2$. The root of the tree is labeled by $\varnothing$.

For example, Fig. 2 shows the one of subtrees from TTP-tree of the frequent patterns mined from the database shown in Fig. 1, where the number after the colon is the support of the pattern, $minsup$ = 3, and $maxgap$ = 2.

To generate all frequent patterns, we first scan the database once to find all frequent 1-patterns and build a projected database for each frequent 1-pattern found. Then, those frequent 1-patterns are added to the level 1 of the TTP-tree.

Next, we recursively extend a frequent k-pattern P ($k \geq 1$) at level k to get its frequent super ($k + 1$)-patterns in a depth-first search manner. To find the frequent super-patterns of P, we scan the projected database of P and find local frequent 1-patterns whose which are connected with P (i.e. the destination of P's k^{th} edge is same as the source location of 1-patterns), and the timespan between P and each frequent 1-pattern found is not greater than $maxgap$. For each frequent 1-pattern found q, we concatenate P with q to form a frequent ($k + 1$)-pattern. During the growth process of TTP-tree, we adopt Lemma 1 to remove the redundant patterns.

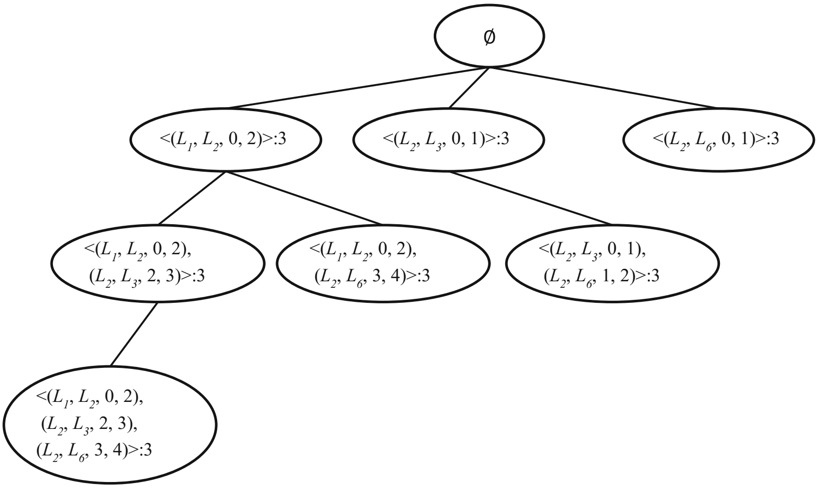

∅
<(L_1, L_2, 0, 2)>:3
<(L_2, L_3, 0, 1)>:3
<(L_2, L_6, 0, 1)>:3
<(L_1, L_2, 0, 2), (L_2, L_3, 2, 3)>:3
<(L_1, L_2, 0, 2), (L_2, L_6, 3, 4)>:3
<(L_2, L_3, 0, 1), (L_2, L_6, 1, 2)>:3
<(L_1, L_2, 0, 2), (L_2, L_3, 2, 3), (L_2, L_6, 3, 4)>:3

Fig. 2. A subtree of TTP-tree.

Lemma 1 (***Same projected database removal***)**.** *If P_1 is a super-pattern of P_2 and both share the same projected database, P_2 can be removed from TTP-tree.*

Proof. Since P_1 is a super-pattern of P_2 and both share the same projected database, every pattern derived from P_2 is contained by a pattern derived from P_1, and both derived patterns have same support. Thus, P_2 and the patterns generated from P_2 should be pruned because they are not closed. ■

Let us take the database shown in Fig. 1 as example, both <(L_1, L_2, 0, 2)> and <(L_1, L_2, 0, 2)> are frequent 1-patterns. Since <(L_1, L_2, 0, 2)> is contained by <(L_1, L_2, 0, 2)> and all of them share the same projected database, we can remove <(L_1, L_2, 0, 1)> from TTP-tree and do not need to grow the 2-patterns from <(L_1, L_2, 0, 1)>.

Let us take the pattern P = <(L_1, L_2, 0, 2)> as an enumerative example. Assume that *minsup* = 3 and *maxgap* = 2. The frequent 1-patterns found from P's projected database are <(L_2, L_3, 2, 3)>, <(L_2, L_6, 3, 4)>. Then we concatenate P with each frequent 1-pattern found to generate frequent 2-patterns. The frequent 2-patterns generated are Q_1 = <(L_1, L_2, 0, 2), (L_2, L_3, 2, 3)>, and Q_2 = <(L_1, L_2, 0, 2), (L_2, L_6, 3, 4)>.

To generate the frequent super-patterns of Q_1 = <(L_1, L_2, 0, 2), (L_2, L_3, 2, 3)>, we scan its projected database, g_1 = <(L_2, L_3, 2, 3), (L_2, L_4, 2, 3), (L_2, L_6, 3, 4), (L_4, L_5, 4, 5), (L_5, L_6, 6, 9)>, g_3 = <(L_2, L_4, 2, 3), (L_2, L_6, 3, 4), (L_7, L_3, 3, 6), (L_4, L_5, 4, 5), (L_5, L_6, 6, 8), (L_3, L_2, 7, 8), (L_1, L_5, 8, 10)>, and find a frequent and non-redundant 1-pattern, <(L_2, L_6, 3, 4)>. Then, we concatenate Q_1 with each frequent 1-pattern found to generate frequent 3-patterns. The frequent 3-pattern Q_3 generated is <($L1$, L_2, 0, 2), (L_2, L_3, 2, 3), (L_2, L_6, 3, 4)>. Similarly, we can recursively mine the other frequent patterns. However, we do not need to generate the super-patterns of Q_2 based on the lemma of *Same projected database removal* since Q_3 is a super-pattern of Q_2 and both share the same projected database.

4.2 The Closure Checking and Pruning Strategies

Based on the method described in Sect. 4.1, we can generate many frequent patterns. However, some generated patterns might not be closed. To reduce to redundant patterns, we leverage the similar idea used in BIDE [13] for closure checking and pruning strategies. When a new frequent pattern is generated, we need to check whether the pattern generated is closed or not.

Lemma 2 (***Forward Redundant Checking Scheme***)**.** *A pattern P is not closed if there exists a frequent 1-pattern e in P's projected database, whose support is equal to P's support, and e is connected with P.*

Proof. If there exists a frequent 1-pattern e in P's projected database whose support is equal to P's support, it means that we can always find another frequent pattern which is formed by concatenating P and e and its support is equal to P's support. Therefore, P must not be closed. ■

We illustrate the *forward redundant checking scheme* as follows. When we generate a new frequent pattern <(L_1, L_2, 0, 2), (L_2, L_3, 2, 3)>, if we find the 1-pattern e = < (L_2, L_6, 3, 4)> which occurs in every graph in the projected database, we can insure that P is not closed. Since P is the sub-pattern of the pattern formed by concatenating P and e, and the patterns formed has the same support as P, P is not closed.

Lemma 3 (***Backward Redundant Checking Scheme***)**.** *A pattern P is not needed to be grown if there exists a frequent 1-pattern e before P, whose support is equal to P's support, the timespan between e and P is not greater than maxgap, and e is connected with P.*

Proof. If there exists a frequent 1-pattern e before P, whose support is equal to P's support and the timespan between e and P is not greater than *maxgap*, it means that we can always find another frequent pattern which is formed by concatenating e and P and its support is equal to P's support. Every pattern generated from P is contained by the pattern generated from concatenating P and e and both patterns have the same support. Thus, P does not need to be grown and it is not closed. ■

Let us explain the *backward redundant checking scheme* strategy by the following example. Assume the pattern P = <(L_2, L_3, 2, 3), (L_2, L_6, 3, 4)> is a frequent pattern. If we find a pattern (say, <(L_1, L_2, 0, 2)> before <(L_2, L_3, 2, 3)>) in every graph containing P and the timespan between <(L_1, L_2, 0, 2)> and P is not greater than *maxgap*, we can conclude that P does not need to be grown. Since there must exist another frequent pattern formed by concatenating <(L_1, L_2, 0, 2)> and P, and the pattern formed shares the same projected database as P, every pattern generated from P is contained by the pattern generated from the pattern formed and both patterns have the same support. Thus, P is not needed to be grown.

4.3 The TTPM-Algorithm

The TTPM-algorithm is shown in Fig. 3. The TTPM-algorithm contains a sub-procedure, TTP-Growth, which is shown in Fig. 4.

```
Algorithm: TSP-algorithm
Input: an input graph database DB, a maximum timespan threshold maxgap,
and a minimum support threshold minsup.
Output: all closed frequent patterns TTP.
Scan the database DB once to find all frequent 1-patterns and build the
projected database for each frequent 1-pattern found;
$TTP=\emptyset$;
for each 1-pattern found P do
  if (P passes the Backward redundant checking scheme ) then
    if (P passes the forward redundant checking scheme ) then
      $TTP=TTP \cup P$;
    end if
    Call TTP-Growth (P, $D|_p$, minsup, maxgap , TTP);
  end if;
end for;
return TTP;
```

Fig. 3. The TTPM-algorithm.

In step 1 of Fig. 3, the TTPM-algorithm first scans the graph database once to find all frequent 1-patterns and build their projected database. For each frequent 1-pattern *P*, if the frequent 1-pattern *P* passes the forward redundant checking scheme, we add *P* to *TTP* in steps 5–7. In step 8, for each frequent 1-pattern found *P*, we call the TTP-Growth sub-procedure to find all closed super-patterns of *P*.

During the mining procedure, we use the *backward redundant checking scheme* strategy to check whether a frequent pattern needs to be grown or not. If this is the case, we will call the TTP-Growth sub-procedure to grow the patterns. The TTP-Growth sub-procedure grows a pattern *P* to find all closed super-patterns of *P*. In the TSP-algorithm, we find all frequent 1-patterns in the projected database, where the timespan between *P* and each frequent 1-pattern found is not greater than *maxgap*. Then we apply the forward and backward redundant checking scheme strategies to check if the frequent patterns generated are closed.

```
Procedure: TTP-Growth
Input: a prefix pattern P, a projected database $DB|_P$, a minimum sup-
port threshold minsup, a maximum timespan threshold maxgap;
Output: all frequent closed patterns TTP.
Scan $DB|_P$ once to find all frequent 1-patterns in the projected data-
base, where  the timespan between P and each frequent 1-pattern found
is not greater than maxgap;
for each frequent 1-pattern found q do
  if (q is connected with any edge of P ) then
    Let $R=P \oplus q$ and build the projected database of R ;
    if (R passes the backward redundant checking scheme) then
      if (R passes the forward redundant checking scheme) then
        $TTP=TTP \cup P$;
      end if;
    Call TTP-Growth(R,$D|_p$, minsup, maxgap, TTP);
    end if;
  end for;
return TTP;
```

Fig. 4. The TTP-Growth sub-procedure.

In the TTP-Growth sub-procedure, from the projected database of *P*, we first find all frequent 1-patterns in step 1. For each frequent 1-pattern found *q*, if *q* is connected with *P*, we concatenate *P* and *q* to form a new frequent pattern *R* and build its projected database in steps 3–4. If *R* passes the forward and backward redundant checking schemes, we add *R* to *TTP* in steps 5–7. Then, in step 9, we recursively call the TTP-Growth sub-procedure to generate the closed frequent patterns.

5 Performance Study

5.1 Settings

We utilize the Gowalla dataset [4] which has been exploited for location-based analysis in several places (such as [11, 12]) to conduct the experiments of time-aware transit pattern mining. The Gowalla dataset contains 6,442,890 check-in records from Feb. 2009 to Oct. 2010. The total number of check-in locations is 1,280,969. We extract two subsets of the check-in data, which correspond to the cities of New York and San Francisco. Some statistics are reported in Table 1.

Table 1. The statistics of the two check-in datasets.

	# of Check-ins	Avg. transaction length	# of days	# of locations
New York	103,174	208.47	428	21,973
San Francisco	187,568	225.45	428	15,406

Since no existing works can tackle the Time-aware Transit Pattern mining problem. We compare our proposed method with the modified Apriori algorithm [1] on execution time. The modified Apriori algorithm generates frequent patterns level by level in a breadth-first search manner. At each level, it combines a frequent *k*-pattern and another frequent *k*-pattern to generate a candidate (*k* + *1*)-pattern. For each candidate (*k* + *1*)-pattern, we scan the database to count its support and check if it is frequent. Note that each candidate (*k* + *1*)-pattern should follow the *maxgap* constraint. The process is repeated until no more frequent patterns can be generated. The modified Apriori algorithm uses only the anti-monotone property to prune the impossible candidates.

5.2 Experimental Results

Figure 5 shows the runtime versus the *minsup* for New York city where the *minsup* varies from 10 % to 50 %. The TTPM-algorithm runs faster than the modified Apriori. As the *minsup* gets smaller, the runtime of the modified Apriori increases sharply but TTPM-algorithm increases slowly. The modified Apriori is more sensitive to the *minsup* than TTPM-algorithm. Since TTPM-algorithm requires only one database scan and remove unnecessary and impossible candidates in projected databases, it is more efficient than the modified Apriori. Similar trend can be found for San Francisco in Fig. 6.

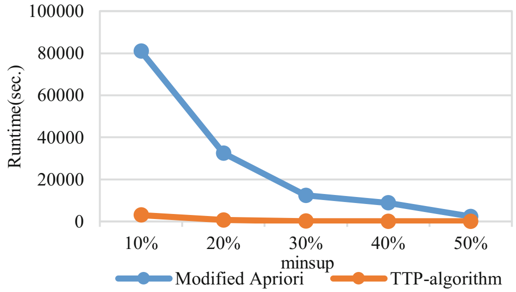


Fig. 5. Runtime versus the *minsup* in New York city.

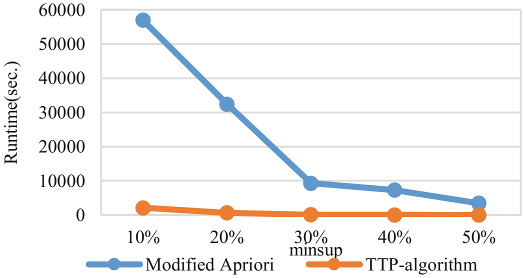


Fig. 6. Runtime versus the *minsup* in San Francisco city.

Figures 7 and 8 show the runtime versus *maxgap* for both algorithms in New York and San Francisco city, where the *maxgap* varies from 1 to 5. As *maxgap* increases, the number of frequent patterns increases. Thus, the runtime of both algorithms increases as the *maxgap* increases. However, as *maxgap* increases, the modified Apriori algorithm would generate many candidates at each level. Therefore, the TTPM-algorithm is more efficient and scalable than the modified Apriori algorithm.

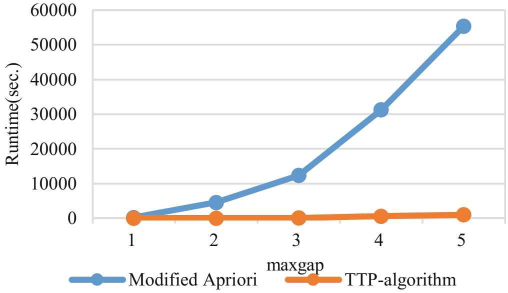


Fig. 7. Runtime versus the *maxgap* in New York city.

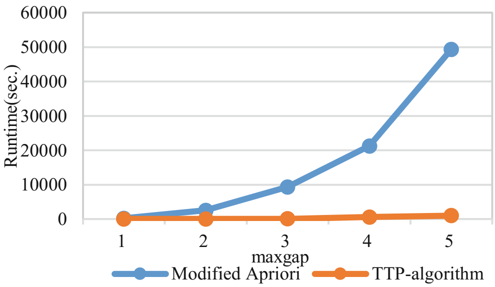


Fig. 8. Runtime versus the *maxgap* in San Francisco city.

6 Conclusion

This paper proposed an efficient algorithm, TTPM-algorithm, to mine the closed time-aware transit patterns in the constructed RTG transaction database. The proposed algorithm recursively performed in a depth-first search manner until no more frequent closed patterns can be found. During the mining process, we use the closure checking scheme and pruning strategies to eliminate impossible and unnecessary candidates. By using the projected database, the TTPM-algorithm only scans the database once and can localize the support counting and candidate pruning in a projected database. Thus, the proposed algorithm can efficiently mine closed patterns in a RTG database. Experimental results show that the TTPM-algorithm outperforms the modified Apriori algorithm in all cases. The found patterns can be used to recommend suitable travel routes and help us have a better understanding of the new city we are the first time to visit. Moreover, the TTPM-algorithm could be used to discover the frequent information diffusion paths, trajectory data, or human interactions. We can use these patterns to predict relationships or human behaviors in the future.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: International Conference on Very Large Data Bases (VLDB), pp. 487–499 (1994)
2. Agrawal, R., Srikant, R.: Mining sequential patterns. In: IEEE International Conference on Data Engineering (ICDE), pp. 3–14 (1995)
3. Ayres, J., Gehrke, J.E., Yiu, T., Flannick, J.: Sequential pattern mining using a bitmap representation. In: ACM SIGMOD International Conference on Knowledge Discovery in Database (SIGMOD), pp. 429–435 (2002)
4. Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 1082–1090 (2011)
5. Inokuchi, A., Washio, T., Motoda, H.: An apriori-based algorithm for mining frequent substructures from graph data. In: European Conference on Principles of Data Mining and Knowledge Discovery (PKDD), pp. 13–23 (2000)

6. Lu, H.C., Lin, C.Y., Tseng, V.S.: Trip-Mine: an efficient trip planning approach with travel time constraints. In: IEEE International Conference on Mobile Data Management (MDM), pp. 162–161 (2011)
7. Kuramochi, M., Karypis, G.: Frequent subgraph discovery. In: IEEE International Conference on Data Mining (ICDM), pp. 313–320 (2001)
8. Leleu, M., Rigotti, C., Boulicaut, J.-F., Euvrard, G.: GO-SPADE: mining sequential patterns over datasets with consecutive repetitions. In: International Conference on Machine Learning and Data Mining, pp. 293–306 (2003)
9. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Discovering frequent closed itemsets for association rules. In: IEEE International Conference on Database Theory (ICDT), pp. 398–416 (1999)
10. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H.: PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth. In: IEEE International Conference on Data Engineering (ICDE), pp. 215–224 (2001)
11. Scellato, S., Noulas, A., Lambiotte, R., Mascolo, C.: Socio-spatial properties of online location-based social networks. In: AAAI International Conference on Weblog and Social Media (ICWSM) (2010)
12. Scellato, S., Noulas, A., Mascolo, C.: Exploiting place features in link prediction on location-based social networks. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 1046–1054 (2011)
13. Wang, J., Han, J., Li, C.: Frequent closed sequence mining without candidate maintenance. IEEE Trans. Knowl. Data Eng. (TKDE) **19**(8), 1042–1056 (2007)
14. Yan, X., Han, J.: gSpan: graph-based substructure pattern mining. In: IEEE International Conference on Data Mining (ICDM), pp. 721–724 (2002)
15. Yan, X., Han, J., Afshar, R.: CloSpan: mining closed sequential patterns in large datasets. In: SIAM International Conference on Data Mining (SDM), pp. 166–177 (2003)
16. Yan, X., Han, J.: CloseGraph: mining closed frequent graph patterns. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 286–295 (2003)
17. Yoon, H., Zheng, Y., Xie, X., Woo, W.: Social itinerary recommendation from user-generated digital trails. Pers. Ubiquit. Comput. **16**, 469–484 (2011)
18. Zaki, M.J.: SPADE: an efficient algorithm for mining frequent sequences. Mach. Learn. **42** (1), 31–60 (2011)
19. Zaki, M.J., Hsiao, C.: Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Trans. Knowl. Data Eng. (TKDE) **17**(4), 462–478 (2005)

Author Index