Wojciech Cellary · Mohamed F. Mokbel
Jianmin Wang · Hua Wang
Rui Zhou · Yanchun Zhang (Eds.)

LNCS 10042

Web Information Systems Engineering – WISE 2016

17th International Conference
Shanghai, China, November 8–10, 2016
Proceedings, Part II

2 Part II

Springer

Lecture Notes in Computer Science 10042

More information about this series at http://www.springer.com/series/7409

Wojciech Cellary · Mohamed F. Mokbel
Jianmin Wang · Hua Wang
Rui Zhou · Yanchun Zhang (Eds.)

Web Information Systems Engineering – WISE 2016

17th International Conference
Shanghai, China, November 8–10, 2016
Proceedings, Part II

Springer

Editors

Wojciech Cellary
Poznań University of Economics
Poznan
Poland

Mohamed F. Mokbel
University of Minnesota
Minneapolis, MN
USA

Jianmin Wang
Tsinghua University
Beijing
China

Hua Wang
Victoria University
Melbourne, VIC
Australia

Rui Zhou
Victoria University
Melbourne, VIC
Australia

Yanchun Zhang
Victoria University
Melbourne, VIC
Australia

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-48742-7 ISBN 978-3-319-48743-4 (eBook)
DOI 10.1007/978-3-319-48743-4

Library of Congress Control Number: 2016955509

LNCS Sublibrary: SL3 – Information Systems and Applications, incl. Internet/Web, and HCI

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Welcome to the proceedings of the 17th International Conference on Web Information Systems Engineering (WISE 2016), held in Shanghai, China, during November 8–10, 2016. The series of WISE conferences aims to provide an international forum for researchers, professionals, and industrial practitioners to share their knowledge in the rapidly growing area of Web technologies, methodologies, and applications. The first WISE event took place in Hong Kong, China (2000). Then the trip continued to Kyoto, Japan (2001); Singapore (2002); Rome, Italy (2003); Brisbane, Australia (2004); New York, USA (2005); Wuhan, China (2006); Nancy, France (2007); Auckland, New Zealand (2008); Poznan, Poland (2009); Hong Kong, China (2010); Sydney, Australia (2011); Paphos, Cyprus (2012); Nanjing, China (2013); Thessaloniki, Greece (2014); Miami, USA (2015); and this year, WISE 2016 was held in Shanghai, China, supported by Fudan University, China.

A total of 233 research papers were submitted to the conference for consideration, and each paper was reviewed by at least three reviewers. Finally, 39 submissions were selected as full papers (with an acceptance rate of 16.7 % approximately), plus 31 as short papers. The research papers cover the areas of social network data analysis, recommender systems, topic modeling, data diversity, data similarity, context-aware recommendation, prediction, big data processing, cloud computing, event detection, data mining, sentiment analysis, ranking in social networks, microblog data analysis, query processing, spatial and temporal data, graph theory and non traditional environments.

In addition to regular and short papers, the WISE 2016 program also featured a special session on Data Quality and Trust in Big Data (QUAT-16) and a medical big data forum.

QUAT is a forum for presenting and discussing novel ideas and solutions related to the problems of exploring, assessing, monitoring, improving, and maintaining the quality of data and trust for big data. It aims to provide researchers in the areas of web technology, e-services, social networking, big data, data processing, trust, and information systems and GIS with a forum for discussing and exchanging their recent research findings and achievements. This year, the QUAT 2016 program featured eight accepted papers on data cleansing, data quality analytics, reliability assessment, and quality of service for domain applications. As the organizers of QUAT 2016, Prof. Deren Chen, Prof. William Song, Dr. Xiaolin Zheng, Dr. Johan Håansson, and Prof. Shaozhong Zhang, we would like to thank all the authors for their enthusiastic high-quality submissions, the reviewers (Program Committee members) for their careful and timely reviews, and the Organizing Committee, Dr. Roger Nyberg, Dr. Zukun Yu, Dr. Xiaofeng Du, and Dr. Xiaoyun Zhao, for their excellent publicity.

The medical big data forum aims to promote the analysis and application of big data in healthcare. Experts and companies related to the domain of big data in healthcare were invited to present their reports in this forum. Many hot research points of big data in healthcare were discussed, including analysis and mining, application and value exploration, interoperability standards, security and privacy protection. The objective of this forum is to provide forward-looking ideas and views for research and application of big data in healthcare, which will promote the development of big data in healthcare, accelerate practical research, and facilitate the innovation and industrial development of mobile healthcare. The forum was organized by Prof. Yan Jia, Prof. Weihong Han, and Prof. Hua Wang.

We also wish to take this opportunity to thank the honorary conference chair, Prof. Maria Orlowska; the general co-chairs, Prof. Hong Mei, Prof. Marek Rusinkiewicz, and Prof. Yanchun Zhang; the program co-chairs, Prof. Wojciech Cellary, Prof. Mohamed F. Mokbel, and Prof. Jianmin Wang; the special area chairs, Prof. Xueqi Cheng, Prof. Yan Jia, and Prof. Jianhua Ma; the workshop co-chairs, Prof. Zhiguo Gong and Prof. Yong Tang; the tutorial and panel chair, Prof. Xuemin Lin; the publication co-chairs, Prof. Hua Wang and Dr. Rui Zhou; the publicity co-chairs, Dr. Jing Yang and Dr. Quan Bai; the website chair, Dr. Rui Zhou; the local arrangements chair, Prof. Shangfei Zhu; the finance co-chairs, Ms. Lanying Zhang and Ms. Irena Dzuteska; the sponsor chair, Dr. Tao Li; and the WISE society representative, Prof. Xiaofang Zhou. The editors and chairs are grateful to Ms. Sudha Subramani and Mr. Sarathkumar Rangarajan for their help in preparing the proceedings and updating the conference website.

We would like to sincerely thank our keynote and invited speakers:

- Professor Maria Orlowska, Fellow of the Australian Academy of Sciences, Vice-President of the Polish-Japanese Institute of Information Technology, former Secretary of State in the Ministry of Science and Higher Education, Poland
- Professor Binxing Fang, academician of CAE (Chinese Academy of Engineering) and the former president of BUPT (Beijing University of Posts and Telecommunications), China
- Dr. Phil Neches, Advisor, Member of National Academy of Engineering, Chairman of Foundation Ventures LLC, founder of Teradata Corporation, USA
- Professor Ramamohanarao (Rao) Kotagiri, Fellow of the Institute of Engineers Australia, Fellow of the Australian Academy Technological Sciences and Engineering, and Fellow of Australian Academy of Science, The University of Melbourne, Australia.

In addition, special thanks are due to the members of the international Program Committee and the external reviewers for the rigorous and robust reviewing process. We are also grateful to Fudan University, China, Victoria University, Australia, and the International WISE Society for supporting this conference. The WISE Organizing Committee is also grateful to the QUAT special session organizers and medical big data forum organizers for their great efforts to help promote web information system research to broader domains.

We expect that the ideas that emerged at WISE 2016 will result in the development of further innovations for the benefit of scientific, industrial, and social communities.

November 2016

Wojciech Cellary
Mohamed F. Mokbel
Jianmin Wang
Hua Wang
Rui Zhou
Yanchun Zhang

Organization

Honorary Conference Chair

Maria Orlowska	Polish-Japanese Institute of Information Technology, Poland

General Co-chairs

Hong Mei	Shanghai Jiao Tong University, China
Marek Rusinkiewicz	New Jersey Institute of Technology, USA
Yanchun Zhang	Victoria University, Australia and Fudan University, China

Program Co-chairs

Wojciech Cellary	Poznań University of Economics, Poland
Mohamed F. Mokbel	University of Minnesota, USA
Jianmin Wang	Tsinghua University, China

Special Area Chairs

Big Data Area Chair

Xueqi Cheng	Chinese Academy of Sciences, China

Medical Big Data Analysis Area Chair

Yan Jia	National University of Defense Technology, China

Transparent Computing and Service Area Chair

Jianhua Ma	Hosei University, Japan

Tutorial and Panel Chair

Xuemin Lin	The University of New South Wales, Australia and East China Normal University, China

Workshop Co-chairs

Zhiguo Gong	University of Macau, Macau, China
Yong Tang	South China Normal University, China

Publication Co-chairs

Hua Wang	Victoria University, Australia
Rui Zhou	Victoria University, Australia

Publicity Co-chairs

Jing Yang	Chinese Academy of Sciences, China
Quan Bai	Auckland University of Technology, New Zealand

Conference Website Chair

Rui Zhou	Victoria University, Australia

Conference Finance Co-chairs

Lanying Zhang	Fudan University, China
Irena Dzuteska	Victoria University, Australia

Local Arrangements Chair

Shangfeng Zhu	Fudan University, China

Sponsorship Chair

Tao Li	Florida International University, USA

Wise Society Representative

Xiaofang Zhou	The University of Queensland, Australia and Soochow University, China

Program Committee

Karl Aberer	EPFL, Switzerland
Imad Afyouni	GIS Technology Innovation Center, Saudi Arabia
Marco Aiello	University of Groningen, The Netherlands
Mohammed Eunus Ali	Bangladesh University of Engineering and Technology, Bangladesh
Toshiyuki Amagasa	University of Tsukuba, Japan
Farnoush Banaei-Kashani	University of Colorado Denver, USA
Jie Bao	Microsoft Research Asia, China
Denilson Barbosa	University of Alberta, Canada
Boualem Benatallah	University of New South Wales, Australia
Azer Bestavros	Boston University, USA

Antonis Bikakis	University College London, UK
Bin Cao	Zhejiang University of Technology, China
Barbara Catania	University of Genoa, Italy
Richard Chbeir	LIUPPA Laboratory, France
Cindy Chen	University of Massachusetts Lowell, USA
Jinchuan Chen	Renmin University of China, China
Jacek Chmielewski	Poznań University of Economics, Poland
Alex Delis	University of Athens, Greece
Schahram Dustar	Vienna University of Technology, Austria
Islam Elgedawy	Middle East Technical University, Turkey
Hicham Elmongui	Alexandria University, Egypt
Marie-Christine Fauvet	Université Grenoble Alpes, France
Yunjun Gao	Zhejiang University, China
Thanaa Ghanem	Metropolitan State University, USA
Claude Godart	Université de Lorraine, France
Daniela Grigori	Laboratoire LAMSADE, Université Paris Dauphine, France
Venkata Gunturi	IIIT-Delhi, India
Hakim Hacid	Bell Labs, USA
Armin Haller	Australian National University, Australia
Mohammad Hammoud	CMU Qatar, Qatar
Tanzima Hashem	Bangladesh University of Engineering and Technology, Bangladesh
Rafiul Hassan	King Fahd University of Petroleum and Minerals, Saudi Arabia
Xiaofeng He	East China Normal University, China
Yuh-Jong Hu	National Chengchi University, Taiwan
Peizhao Hu	Rochester Institute of Technology, USA
Jianbin Huang	Xidian University, China
Marta Indulska	University of Queensland, Australia
Yoshiharu Ishikawa	Nagoya University, Japan
Adam Jatowt	Kyoto University, Japan
Yan Jia	National University of Defense Technology, China
Lili Jiang	Max Planck Institute for Informatics, Germany
Wei Jiang	Missouri University of Science and Technology, USA
Peiquan Jin	University of Science and Technology of China, China
Byeong Ho Kang	University of Tasmania, Australia
Raymond Lau	City University of Hong Kong, Hong Kong, SAR China
Dan Lin	Missouri University of Science and Technology, USA
Shuai Ma	Beihang University, China
Murali Mani	University of Michigan-Flint, USA
Natwar Modani	Adobe Research, India
Mikolaj Morzy	Poznań University of Technology, Poland
Wilfred Ng	Hong Kong University of Science and Technology, Hong Kong, SAR China

Kjetil Nørvåg	Norwegian University of Science and Technology, Norway
Mitsunori Ogihara	University of Miami, USA
George Pallis	University of Cyprus, Cyprus
Wen-Chih Peng	National Chiao Tung University, Taiwan
Olivier Pivert	ENSSAT, France
Tieyun Qian	Wuhan University, China
Jarogniew Rykowski	Poznań University of Economics, Poland
Yucel Saygin	Sabanci University, Turkey
Wei Shen	Nankai University, China
John Shepherd	University of New South Wales, Australia
Lawrence Si	University of Macau, Macau, SAR China
Dandan Song	Beijing Institute of Technology, China
Shaoxu Song	Tsinghua University, China
Reima Suomi	University of Turku, Finland
Stefan Tai	Karlsruhe Institute of Technology, Germany
Dimitri Theodoratos	New Jersey Institute of Technology, USA
Yicheng Tu	University of South Florida, USA
Xiaojun Wan	Peking University, China
Hua Wang	Victoria University, Australia
Junhu Wang	Griffith University, Australia
De Wang	Google, USA
Ingmar Weber	Qatar Computing Research Institute, Qatar
Adam Wojtowicz	Poznań University of Economics, Poland
Jei-Zheng Wu	Soochow University, Taiwan
Takehiro Yamamoto	Kyoto University, Japan
Hayato Yamana	Waseda University, Japan
Yanfang Ye	West Virginia University, USA
Hongzhi Yin	The University of Queensland, Australia
Tetsuya Yoshida	Nara Women's University, Japan
Ge Yu	Northeastern University, China
Jeffrey Xu Yu	Chinese University of Hong Kong, Hong Kong, SAR China
Qi Zhang	Fudan University, China
Xiangmin Zhou	RMIT University, Australia
Xingquan Zhu	Florida Atlantic University, USA

QUAT General Co-chairs

Deren Chen	Zhejiang University, China
William Song	Dalarna University, Sweden

QUAT Program Committee Co-chairs

Xiaolin Zheng	Zhejiang University, China
Johan Håkansson	Dalarna University, Sweden
Shaozhong Zhang	Zhejiang Wanli University, China

QUAT Organizing Committee Co-chairs

Roger G. Nyberg	Dalarna University, Sweden
Zukun Yu	Britich Telecom, UK
Xiaofeng Du	British Telecom, UK
Xiaoyun Zhao	Dalarna University, Sweden

QUAT Program Committee

Adriana Marotta	Universidad de la República, Uruguay
Anders Avdic	Dalarna University, Sweden
Fei Chiang	McMaster University, Canada
Hasan Fleyeh	Dalarna University, Sweden
Jacky Keung	City University of Hong Kong, Hong Kong, SAR China
Jun Hu	Nanchang University, China
Preben Hansen	Stockholm University, Sweden
Rajeev Agrawal	North Carolina A&T State University, USA
Sheng Zhang	Nanchang Hangkong University, China
Yuansheng Zhong	Jiangxi University of Finance and Economics, China
Yuhao Wang	Nanchang University, China

QUAT Sponsors

Complex Systems & Microdata Analysis, Dalarna University, Sweden
E-Service Research Center, Zhejiang University, China

Contents – Part II

Sentiment Analysis

Ranking in Social Networks

Microblog Data Analysis

Query Processing

Spatial and Temporal Data

Graph Theory

Non-traditional Environments

Special Session on Data Quality and Trust in Big Data

Contents – Part I

Social Network Data Analysis

Recommender Systems

Topic Modeling

Data Diversity

Data Similarity

Context-Aware Recommendation

Prediction

Big Data Processing

Cloud Computing

Event Detection

Data Mining

Sentiment Analysis

Dynamic Topic-Based Sentiment Analysis of Large-Scale Online News

Peng Liu(✉), Jon Atle Gulla, and Lemei Zhang

Department of Computer and Information Science, NTNU, Trondheim, Norway
{peng.liu,jon.atle.gulla,lemei.zhang}@idi.ntnu.no

Abstract. Many of today's online news websites and aggregator apps have enabled users to publish their opinions without respect to time and place. Existing works on topic-based sentiment analysis of product reviews cannot be applied to online news directly because of the following two reasons: (1) The dynamic nature of news streams require the topic and sentiment analysis model also to be dynamically updated. (2) The user interactions among news comments can easily lead to inaccurate topic and sentiment extraction. In this paper, we propose a novel probabilistic generative model (DTSA) to extract topics and the specified sentiments from news streams and analyze their evolution over time simultaneously. DTSA incorporates a multiple timescale model into a generative topic model. Additionally, we further consider the links among news comments to avoid the error caused by user interactions. Finally, we derive distributed online inference procedures to update the model with newly arrived data and show the effectiveness of our proposed model on real-world data sets.

Keywords: Topic-based sentiment analysis · Topic model · User interaction · Online inference

1 Introduction

With the growing popularity of both the social media and mobile news apps, an increasingly amount of significant information concerning user opinions and sentiments is being stored online. As important platforms used to describe events happening around the world, online news and comments are the efficient means of conveying positive or negative emotions underlying an opinion and also communicating an affective state, such as happiness, fearfulness, or surprise. It is valuable to extract topics as well as sentimental information from these texts. The governments can detect public sentiments toward policies and emergencies and give feedback in time. The marketers are able to acquire knowledge about the public sentiment environment which supports further analysis and decisions. However, the analysis is impossible to complete manually due to the huge amount of data, and the unstructured data increases the difficulty of machine analysis.

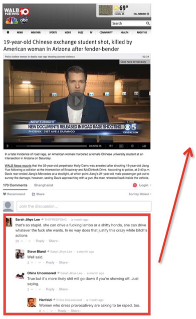
Most earlier studies [1–3] embrace topics or domains into sentiment analysis model, to improve the accuracy of sentiment classification. To a large extent,

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 3–18, 2016.
DOI: 10.1007/978-3-319-48743-4_1

it is due to the tightly reliance on domains or topics of sentiment description. The same word in different topics may convey various sentiment polarities. For instance, the word "offensive" is used as positive orientation in the phrase "offensive player" when discussing sports news, whereas it also has negative orientation when used in the phrase "offensive behaviour" referring to political news comments. Thus, sentiment analysis based on topic or domain has far-reaching significance.

In recent years, among the many researches on the approaches to extract topic-based sentiments, most works have focused on analyzing product comments, which are very different from the comments on news and events [4]. More specifically, current studies assume that words in documents have static co-occurrence patterns, which may not be suitable for the task of capturing topic and sentiment shifts in a time-variant data corpus. What is more, the most popular topic models for sentiment analysis rely on batch mode learning which assumes that the training data are all available prior to model learning. When fitting large-scale news streams, the time and memory costs of such approaches will scale linearly with the number of documents analyzed. In addition, many algorithms regard comments as independent individuals, ignoring their connections. In fact, the socialized characteristic of the media platform makes it easier for users to interact with each other, which will result in more connections.

To have a better understanding of user interaction, we list some real comments with interactions of the WALB News website and their corresponding polarities and types in Fig. 1. The first comment shows a negative sentiment towards the shooting news. The second comment agrees with the first comment's opinion using positive expressions whereas the third person has a little disagreement with the first one. The last comment is based on the previous critiques. In such a situation, we find some drawbacks in the existing methods. First, for example, in the comment "Well said", the existing methods cannot extract the corresponding topics unless considering the interaction with the original news comment. Second, the normal sentiment polarities of positive and negative cannot precisely describe the sentiment polarities of news comments between



19-year-old Chinese exchange student shot, killed by American woman in Arizona after fender-bender

Police believe woman in deadly road rage shooting planned violence 02:34

In a fatal incidence of road rage, an American woman murdered a female Chinese university student at an intersection in Arizona on Saturday.

WALB News reports that the 32-year-old perpetrator Holly Davis was arrested after shooting 19-year-old Jiang Yue following a collision at the intersection of Broadway and McClintock Drive. According to police, at 2:40 p.m. Davis rear-ended Jiang's Mercedes at a stoplight, at which point Jiang's 21-year-old male passenger got out to survey the damage; however, seeing Davis approaching with a gun, the man retreated back inside the vehicle.

Sarah Jihye Lee → THEFREDFONG · a month ago
that's so stupid. she can drive a fucking lambo or a shitty honda, she can drive whatever the fuck she wants. In no way does that justify this crazy white bitch's actions
29 · Reply · Share ›

Steve Bland → Sarah Jihye Lee · a month ago
Well said.
3 · Reply · Share ›

China Uncensored → Sarah Jihye Lee · a month ago
True but it's more likely shit will go down if you're showing off. Just saying.
2 · Reply · Share ›

Hanfeizi → China Uncensored · a month ago
Women who dress provocatively are asking to be raped, too.
8 · Reply · Share ›

News Comments	Type (comment/response)	Polarity of News (positive/negative)	Polarity of Existing Methods (positive/negative)
That's so stupid. She can drive a fucking lambo or a shitty honda, she can drive whatever the fuck she wants. In no way does that justify this crazy white bitch's actions.	comment	negative	negative
Well said.	response	negative	positive
True but it's more likely shit will go down if you're showing off. Just saying.	response	negative	negative
Women who dress provocatively are asking to be raped, too.	response	negative	negative

Fig. 1. News comments and the interactions between them.

interactions, so the sentiment classification results will be inaccurate using existing methods. Therefore, user interaction affects both the extraction of topics and sentiments, which renders existing methods less useful.

In this paper, we propose a dynamic topic-based sentiment analysis model (DTSA) which is capable of extracting topics and topic-specific sentiments from the online news comment and tracking their evolution over time simultaneously. The DTSA model incorporates the links among new comments to avoid the error caused by user interactions. To efficiently handle streaming data, we derive online inference procedures based on a stochastic Expectation Maximization (EM) algorithm, in which the model is sequentially updated using newly arrived data and the parameters of the previously estimated model. We applied our model to several real data sets and the experimental results demonstrate promising and reasonable performance of our approach.

This paper makes the following contributions:

– It proposes a DTSA model where the generation of current sentiment-topic-word distributions are influenced by the multiple timescale word distributions at the previous epoch. Considering both the long-timescale dependency and the short-timescale dependency improves the robustness of the model.
– Two special sentiments which represent the transformation of user sentiments–approval and disapproval are introduced to model the links among news comments, which could improve the accuracy of topic-based sentiment classification.
– The proposed DTSA approach adopts a distributed online inference procedure to update the model with newly arrived data, which can be generalized to perform dynamic topic-based sentiment analysis on other large-scale social media streams.

The remainder of the paper is organized as follows. Section 2 introduces the related work. In Sect. 3, we present our new model. We describe the data sets, experiment settings and the prior information we use in Sect. 4. Section 5 shows our experiment results. Finally, we present the conclusions and future work in Sect. 6.

2 Related Work

Although much work has been done in detecting topics [5–7], these works mainly focused on discovering and analyzing topics of documents alone, without any analysis of sentiments in the text, which limit the usefulness of the mining results. Other works [8,9] addressed the problem of sentiment detection at various levels (i.e. from word/phrase level, to sentence and document level). However, none of them can model the mixture of topics and sentiment classification, which again makes the results less informative to users.

Some of the recent works [1–3] have been aware of this limitation and tried to capture sentiments and the mixture of topics simultaneously. Lin and He [1] introduced sentiment polarities into topic modeling and presented a model called

JST which can extract the mixture of aspects and different sentiment polarities for products and services. Aspect-and-Sentiment Unification Model (ASUM) [2] and Sentiment-Topic model with Decomposed Prior (STDP) [3] are all based on LDA, which extract sentiments about topics in a static way without consideration of the dynamic nature of documents. Besides, these methods do not take into account the sentiment polarity transformation caused by user interactions.

In recent years, there has been a surge of interest in developing topic models to explore topic evolutions over time. The continuous time dynamic topic model (cDTM) [10] used Brownian motion to model the latent topics through a sequential collection of documents. [12] proposed online multiscale dynamic topic models (OMDT) which could trace the topic evolution with multiple timescales. It was on the basis of the Dirichlet-multinomial framework by assuming that current topic-specific distributions over words were generated based on the multiscale word distributions of the previous epoch. Wang et al. [13] proposed a Temporal-LDA or TMLDA method to mine streams of social text such as the Twitter stream for an author, by modeling the topics and topic transitions that naturally arised in such data. Different from the work of [10], it focused more on learning the relationship among topics.

None of the aforementioned models take into account time-aware topic–sentiment analysis. Mohamed et al. [14] proposed an LDA based topic model for analyzing topic-sentiment evolution over time by modeling time jointly with topic and sentiments, and derived inference algorithm based on Gibbs Sampling process. However, this time-aware topic-sentiment (TTS) model could not consider adjusting model parameters in realtime and process online news streams. [15] presented probabilistic model called topic sentiment trend model (TSTM), based on probabilistic latent semantic analysis (PLSA) model. Thus it exists the problems of inferencing on new documents and overfitting the data.

Our model DTSA is partly inspired by the previously proposed multiscale topic models [12] and explores the generation process of online comments, considering the co-effects caused by user interactions and the time factor for the first time. The results show that the DTSA model makes a significant improvement on both topics and topic-specific sentiments extraction.

3 The DTSA Model

In this section, we propose a novel dynamic topic-based sentiment analysis model (DTSA) for large-scale online news. Firstly, the problem is defined, including the relevant general terms and notations. Then a multiple timescale model and a graphical model are presented in detail. Finally, we describe the estimation and prediction of parameters.

3.1 Problem Definition

For convenience of describing the graphical model, we here define the following terms and notations:

In a time-stamped news comments collection, we assume comments are sorted in the ascending order of their time stamps. At each epoch t where the time period for an epoch can be set arbitrarily at an hour, a day, or a year. A stream of comments $C^t = \{c_1^t, c_2^t, c_3^t, ..., c_D^t\}$ are received with their order of publication time stamps preserved.

In C^t, D is the number of comments, K is the number of topics, S_1 is the number of normal sentiments (positive and negative), S_2 is the number of special sentiments (approval and disapproval), and $M = S_1 + S_2$ is the total number of sentiments. n_d^s is the number of sentiment words in comment d and n_d^o is the number of topic words in comment d. There are K topic models $\varphi_{z=1...K}^o$ which denotes the multinomial distribution of words specific to topic z. For each topic z, there are S_1 topic-specific normal sentiment models $\varphi_{l=1...S_1,z}^n$, which denotes the multinomial distribution of words specific to normal sentiment label l and topic z. There are S_2 special sentiment models $\varphi_{m=1...S_2}^s$, which is the multinomial distribution of words specific to special sentiment label m. The variable θ denotes the distribution of topics in comment d, the variable π denotes the distribution of sentiments in comment d. Let $d^{'}$ be the comment that d interacts with, then the variables $\theta^{'}$ and $\pi^{'}$ denote the distribution of topics and sentiments in comment $d^{'}$.

In particular, we define an evolutionary matrix of topic z and sentiment label l, $E_{l,z}^t$, where each column is the word distribution of topic z and sentiment label l, $\sigma_{l,z,s}^t$, generated for comments received within the time slice specified by s. We then attach a vector of weights $\mu_{l,z}^t = \{\mu_{l,z,0}^t, \mu_{l,z,0}^t, \mu_{l,z,0}^t, ..., \mu_{l,z,s}^t\}$, each of which determines the contribution of time slice s in computing the priors of $\beta_{l,z}^t$.

The Key Task of Dynamic Topic-based Sentiment Analysis (DTSA) is to estimate the model parameters σ^t, μ^t, θ^t, π, φ^o, φ^n and φ^s using a stochastic EM algorithm, then to extract topics and topic-specific sentiments of the online news and analyze their evolution over time simultaneously. Table 1 summarizes the notations of frequently used variables.

3.2 Multiple Timescale Model

Following the previous work [12], we could account for the influence of the past at different timescales to the current epoch. For example, we set time slice s equivalent to 2^{S-1} epochs. Hence, if S = 3, we would consider three previous sentiment-topic-word distributions where the first distribution is between epoch t − 4 and t − 1, the second distribution is between epoch t − 2 and t − 1, and the third one is at epoch t − 1. This would allow taking into consideration of previous long and short timescale distributions. However, this model would take more time and memory spaces and effective algorithm needs to be performed in order to reduce time/memory complexity.

Figure 2 illustrates the relationship among μ, E and β when the number of historical time slices accounted for is set to 3. Here, $\sigma_{l,z,s}^t$, $s \in 1...3$ is the historical word distribution of topic z and sentiment label l within the time slice specified by s. As a form of smoothing to avoid the zero probability problem for unseen words, we set $\sigma_{l,z,0}^t$ for the current epoch as the uniform

Table 1. Notations used in the paper.

Symbol	Description
t	The index of timestamp
K	Number of topics
D	Number of comments
S_1	Number of normal sentiments (positive and negative)
S_2	Number of special sentiments (approval and disapproval)
M	The total number of sentiments
N_d^s	Number of sentiment words in comment d
N_d^t	Number of topic words in comment d
γ	Symmetric prior for sentiment labels
α	The prior for the topic distribution
β	The prior for the word distribution conditioned on sentiment labels and topics
φ_z^o	The multinomial distribution of words specific to topic z
$\varphi_{l,z}^n$	The multinomial distribution of words specific to normal sentiment label l and topic z
φ_m^s	The multinomial distribution of words specific to special sentiment label m
λ	The word prior for sentiment polarity information
θ	The distribution of topics in comment d
π	The distribution of sentiments in comment d
$\theta^{'}$	The distribution of topics in the comment that d interacts with
$\pi^{'}$	The distribution of sentiments in the comment that d interacts with
$E_{l,z}^t$	Evolutionary matrix of sentiment label l and topic z at epoch t
$\mu_{l,z}^t$	Weight vector which determines the contribution of time slice s in computing the priors of $\beta_{l,z}^t$
$\sigma_{l,z,s}^t$	The multinomial word distribution of sentiment label l and topic z with time slice s at epoch t

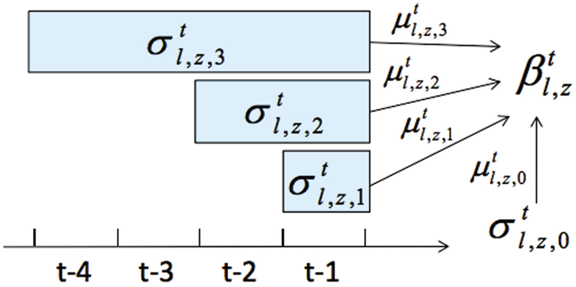


Fig. 2. The relationship among μ, E and β.

distribution where each element takes the value of 1/(vocabulary size). The evolutionary matrix $E_{l,z}^t = \{\sigma_{l,z,0}^t, \sigma_{l,z,1}^t, \sigma_{l,z,2}^t, \sigma_{l,z,3}^t\}$, and the weight matrix $\mu_{l,z}^t = \{\mu_{l,z,0}^t, \mu_{l,z,1}^t, \mu_{l,z,2}^t, \mu_{l,z,3}^t\}^T$. The Dirichlet prior for sentiment-topic-word distributions at epoch t is $\beta_{l,z}^t = \mu_{l,z}^t E_{l,z}^t$.

3.3 Graphical Model

According to the real-world observation, we give two assumptions on sentiments as follow: (1) The sentiments of a comment do not exist independently, but depend on the comment it replies to and their relationship. (2) News comments can be divided into the reply comments and the original comments. The reply often omits the topic information, because it has the same topic with the original. We call this characteristic of user interaction "Topic Consistency".

The graphical representation of DTSA is shown in Fig. 3. The parameter definitions are shown in Table 1.

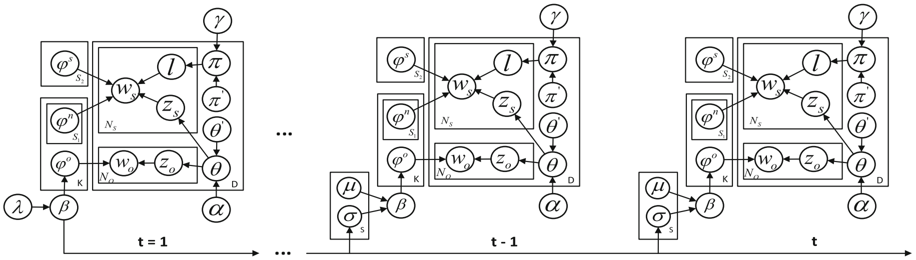


Fig. 3. DTSA model.

Assuming we have already calculated the evolutionary parameters $\{E_{l,z}^t, \mu_{l,z}^t\}$ for the current epoch t, the formal generative process of DTSA model as shown in Fig. 3 at epoch t is given as follows:

1. For each normal sentiment $l \in \{1, ..., S_1\}$:
   i. For each topic $z \in \{1, ..., K\}$:
      Compute $\beta_{l,z}^t = \mu_{l,z}^t E_{l,z}^t$
2. For each topic $z \in \{1, ..., K\}$:
   i. Choose a distribution $\varphi_z^o \sim Dir(\beta_z^o)$
   ii. For each normal sentiment $l \in \{1, ..., S_1\}$
      Choose a distribution $\varphi_{l,z}^n \sim Dir(\beta_{l,z}^n)$
3. For each special sentiment $m \in \{1, ..., S_2\}$:
   Choose a distribution $\varphi_m^s \sim Dir(\beta_m^s)$
4. For each comment $d \in \{1, ..., D\}$:
   i. Choose a distribution $\theta_{temp} \sim Dir(\alpha)$:
      Create a new distribution θ_d by combining θ_{temp} and $\theta_{d'}^{'}$
   ii. Choose a distribution $\pi_{temp} \sim Dir(\gamma)$:
      Create a new distribution π_d by combining π_{temp} and $\pi_{d'}^{'}$
   iii. For each topic word $w_{d,i}^o$ where $i \in \{1, ..., n_d^o\}$:
      (a) Choose a topic $z_i^o \sim Mult(\theta_d)$
      (b) Choose a word $w_{d,i}^o$ from the distribution φ^o over words defined by the topic z_i^o.

iv. For each sentiment word $w^s_{d,j}$ where $j \in \{1, ..., n^s_d\}$:
   (a) Choose a topic $z^s_j \sim Mult(\theta_d)$
   (b) Choose a sentiment label $l_j \sim Mult(\pi_d)$
   (c) If l_j is a normal sentiment, choose a sentiment word $w^s_{d,j}$ from the distribution φ^n over words defined by the topic z^s_j and sentiment l_j. Otherwise, choose a special sentiment word $w^s_{d,j}$ from the distribution φ^s over words defined by the sentiment m_j.

In the proposed model, we divide the words into topic words and sentiment words. A sentiment lexicon and POS tagging are used to identify the sentiment words. There are two kinds of sentiments in the model-normal and special ones. The normal sentiments are topic-sensitive, where users use different words to express the same sentiment in different topics. However, the special sentiments are not topic-sensitive. According to [16], there are some patterns in approval and disapproval. Therefore, we choose the distributions of all k topics for each normal sentiment S^n, but only one distribution is chosen for each special sentiment S^s.

The topics and sentiments of the comment are affected by the comment a user interacts with, so we introduce the topics distribution $\theta^{'}$ and sentiments distribution $\pi^{'}$ of the interacted comment to reflect this effect. Intuitively, we expect the two distributions θ and $\theta^{'}$ are linear correlation, where $\theta = p\theta^{'} + (1 - p)\theta_{temp}$. The greater p value means a better topic consistency, which depends on the data set. We also expect $\pi = q\pi^{'} + (1 - q)\pi_{temp}$. Approximately, a larger q represents more weight on user interactions. The setting for p and q was determined empirically.

3.4 Online Inference

We use a stochastic EM algorithm to sequentially update the model parameters at each epoch using the newly arrived data and the parameters of the previously estimated model. At each EM iteration, we infer latent sentiment labels and topics using the collapsed Gibbs sampling and estimate the hyperparameters using maximum likelihood [17]. Due to the space limit, we leave out the derivation details and only show the sampling formulas.

Model Parameters Estimation. The sampling formulas of model parameters $\theta^t, \pi^t, \varphi^o_t, \varphi^n_t$ and φ^s_t at epoch t given the evolutionary parameters E^t, μ^t are follow:

$$\theta^t_{d,k} = \frac{N^o_{d,k,t} + N^s_{d,k,t} + \alpha^t_k}{\sum_{k=1}^{K}(N^o_{d,k,t} + N^s_{d,k,t} + \alpha^t_k)} \tag{1}$$

$$\pi^t_{d,m} = \frac{N^s_{d,m,t} + \gamma^t_m}{\sum_{m=1}^{M}(N^s_{d,m,t} + \gamma^t_m)} \tag{2}$$

$$\varphi^o_{k,v,t} = \frac{N^o_{k,v,t} + \sum_S \mu^t_{k,s,v}\sigma^t_{k,s,v}}{\sum_{v=1}^{V}(N^o_{k,v,t} + \sum_S \mu^t_{k,s,v}\sigma^t_{k,s,v})} \tag{3}$$

$$\varphi^n_{k,m,v,t} = \frac{N^n_{k,m,v,t} + \sum_S \mu^t_{k,m,s,v}\sigma^t_{k,m,s,v}}{\sum_{v=1}^{V}(N^n_{k,m,v,t} + \sum_S \mu^t_{k,m,s,v}\sigma^t_{k,m,s,v})} \tag{4}$$

$$\varphi^s_{m,v,t} = \frac{N^s_{m,v,t} + \sum_S \mu^t_{m,s,v}\sigma^t_{m,s,v}}{\sum_{v=1}^{V}(N^s_{m,v,t} + \sum_S \mu^t_{m,s,v}\sigma^t_{m,s,v})} \tag{5}$$

where $N^o_{d,k,t}$ is the number of topic words assigned to topic k in document d at epoch t. $N^s_{d,k,t}$ is the number of sentiment words assigned to topic k in document d at epoch t. $N^s_{d,m,t}$ is the number of sentiment words assigned to sentiment m in document d at epoch t. Other variables containing N are defined similarly.

Evolutionary Parameters Estimation. There are two sets of evolutionary parameters to be estimated, the weight parameters μ and the evolutionary matrix E. The update formulas are:

$$(\mu^t_{k,m,s})^{new} \leftarrow \frac{\mu^t_{k,m,s}\sum_v \sigma^t_{k,m,s,v}A}{B} \tag{6}$$

where $A = \psi(N^t_{k,m,v} + \sum_{s'}\mu^t_{k,m,s'}\sigma^t_{k,m,s',v}) - \psi(\sum_{s'}\mu^t_{k,m,s'}\sigma^t_{k,m,s',v})$ and $B = \psi(N^t_{k,m} + \sum_{s'}\mu^t_{k,m,s'}) - \psi(\sum_{s'}\mu^t_{k,m,s'})$, $N^t_{k,m,v}$ is the number of times word v assigned to sentiment label m and topic k at epoch t, $N^t_{k,m} = \sum_v N^t_{k,m,v}$.

The evolutionary matrix E^t accounts for the historical word distributions at different time slices. The derivation of E^t therefore requires the estimation of each of its elements, $\sigma^t_{k,m,s,v}$, the word distribution in topic k and sentiment label m at time slice s, which can be calculated as follows:

$$\sigma^t_{k,m,s,v} = \frac{C^t_{k,m,s,v}}{\sum_v C^t_{k,m,s,v}} \tag{7}$$

where $C^t_{k,m,s,v}$ is the expected number of times word v is assigned to sentiment label m and topic k at time slice s. For the Multi-scale model, a time slice s might consist of several epochs. Therefore, $C^t_{k,m,s,v}$ is calculated by accumulating the count $N^{t'}_{k,m,v}$ over several epochs. The formula for computing $C^t_{k,m,s,v}$ is $C^t_{k,m,s,v} = \sum_{t'=t-2^{s-1}}^{t-1} N^{t'}_{k,m,v}$.

Distributed Model Training. To handle large scale data sets, we design a parallel training program for DTSA model on Hadoop, which is a Java-based open source distributed computing framework. Hadoop implemented the MapReduce framework proposed by Jeffrey et al. [18], and it can effectively handle a large amount of data. In Hadoop, all data are stored as key-value pairs. For our proposed model training program, the key is document id, and the value is the words and sentiments in the comment with their corresponding latent topics. The global model parameters include the Dirichlet prior α^t, γ^t, the weight parameter μ^t and the element of evolutionary matrix σ^t. Initially, a comment set is randomly split into N equal parts for N parallel executing processes. In the Map stage, every process loads the global model parameters from the last iteration, and uses them to sample the comments in its own part. The posterior distribution of hidden variables θ^t, π^t, φ_t^o, φ_t^n and φ_t^s are computed. In the Reduce stage, the posterior distribution θ^t, π^t, φ_t^o, φ_t^n and φ_t^s from all processes are aggregated to generate a new version of global model parameters.

4 Experimental Setup

We evaluate our proposed model on two kinds of datasets: news and twitter. For news datasets, we crawl the comments of four hot news events occurred from February 2014 to April 2014 using the Guardian Open Platform API[1]. (1) MH370 event: Malaysia airlines MH370 B777-200ER loses contact with air traffic control. (2) Crimea event: Russia dispatches troops to Crimea. (3) Sochi event: Sochi 2014 Winter Olympics are held successfully. (4) India event: India holds the largest president election ever. In order to evaluate our model's generality, we also crawl the tweets of Facebook events occurred on February 2014 from Twitter Search API[2]. Facebook event: Facebook buys WhatsApp for 19 Billion US Dollars. Each dataset contains the comments interacted with other comments by reply. Detail statistics of the datasets and sentiment distribution are shown in Table 2.

Table 2. Some statistics of the datasets and sentiment distribution.

Dataset	MH370	Crimea	Sochi	India	Facebook
Documents	351041	46722	405000	289900	101900
Tokens	2565490	382419	3518923	2359761	1026950
# of positive/negative documents	7/12	3/8	7/2	5/4	4/7

DTSA is an unsupervised model. As preprocessing, we first perform stemming and remove stopwords. Then we use Stanford POS Tagger[3] to tag the comments. In prior information, we use the sentiment lexicon SentiWordNet[4], containing 2290 positive and 4800 negative words with score over 0.6, as normal sentiment. Words contained in the sentiment lexicon are automatically labeled as sentiment words. For special sentiment, we use some seed words as prior information for approval, such as “praise”, “agree”, “support”, and we use the discourse markers and swear words as prior information for disapproval, such as “what?”, “nonsense” [16]. Other words, which are not labeled as normal/special sentiment words, are regarded as topic words. To quantitatively evaluate our model,

[1] http://open-platform.theguardian.com/.
[2] http://apiwiki.twitter.com/.
[3] http://nlp.stanford.edu/software/tagger.shtml.
[4] http://sentiwordnet.isti.cnr.it/.

we randomly select 500 comments from five datasets separately, and manually label each word as topic, normal and special sentiment word.

In our experiments, the unit epoch is set to daily. The number of topics K is set to be 20, the number of normal sentiments S_1 is set to be 2, the number of special sentiments S_2 is set to be 2. We set the Gibbs sampling iterations to be 5000. Following [6], we fix $\alpha = 50/K$, $\gamma = 50/(S_1 + S_2)$.

5 Experiments

In this section, we evaluate the performances of our proposed models with three experiments. In the first experiment, we show the topics and topic-specific sentiments extracted by DTSA with some qualitative analysis. The second experiment evaluates the computational time of our models. In the third experiment, we apply a document-level sentiment classification task to compare our models with several baselines.

5.1 Qualitative Results

In Table 3 we show the evolution of topics and topic-specific sentiments identified by the DTSA model with the number of time slices set to 4. Due to space limit, we only take an example of news comments on the MH370 event. For each topic, we list the top 5 topic words and the related sentiment words.

We can see that DTSA can extract topics and topic-based sentiments well. For example, the topic words are “MH370” and “disappeared” while the specific negative sentiment words are “painful” and “cruel”. We also notice that the evolution of topics is well consistent with the actual news stories in real world. In addition, one improvement of the proposed model is that DTSA could automatically adjust the polarity of sentiment words. For example, in Epoch 9, the word

Table 3. News MH370 lose contact.

Time	Epoch 5	Epoch 6	Epoch 7	Epoch 8	Epoch 9	Epoch 10
Topic	MH370 Malaysia flight missing search	MH370 disappeared passenger plane terrorism	officials MH370 search scientists Australian	MH370 military sea evidence found	Boeing fuel MH370 died people	MH370 aircraft underwater Sumatra search
Senti_Positive	hopeful prospective extreme promising optimistic	optimistic hopeful cheerful confident huge	advanced optimistic sophisticated powerful support	support hopeful sustained sufficient powerful	hopeful promising likely confident high	prospective optimistic strong huge advanced
Senti_Negative	sad bad dangerous unbelievable tragic	cruel sorrily ruefully painful tearing	miserable painful tearing sorrily fierce	tragic unwilling hate cruel sorrowful	hopeless sad despairing suffering believe	hopeless ruefully painful sad misery

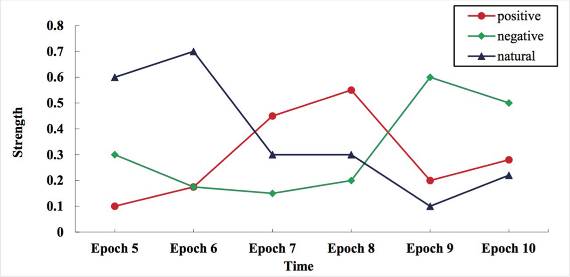


Fig. 4. Sentiment dynamics of MH370 event.

"believe" becomes negative while it is positive in lexicon. In the comment "So what? We just believe they are alive!", "believe" should have labeled this comment positive, but the prior information "what?" makes this comment labeled as disapproval. And because this comment is a reply to a comment which approves of the news topic, we change the sentiment distribution of this comments to disapprove of the topic, which makes "believe" becomes negative words.

In Fig. 4, we plot and compare the topic life cycle and its sentiment dynamics on MH370 event, where the strength distribution of a sentiment l in document d associated with the topic z, over the comment set C^t in each epoch t is calculated as:

$$P(z,l) = \frac{1}{|C^t|} \sum_{d \in C^t} P(z|l,d)P(l|d) \tag{8}$$

From Fig. 4, we can see that in the first 2 days, the neutral sentiment dominates the opinions, for everyone talks about the facts during that time. However, the positive sentiment rises obviously over the next 2 days, reaching the peak at day 4, since the search and rescue operations. After that, the negative sentiment shoots up for 24 h, peaking at day 5. This is mainly because the Boeing 777 has run out of fuel and passengers have little chance of survival. All these results show that DTSA is effective to extract topics and topic-specific sentiments.

5.2 Evaluation of Computational Time

In order to evaluate the effectiveness of DTSA in modelling dynamics, we compare the computational time of the DTSA model with the non-dynamic version of LDA [5] and JST [17], namely, LDA-one, JST-one, and JST-all. LDA-one and JST-one only use the training data in the current epoch whereas JST-all uses all the past data for model learning.

According to the previous work [10,11], we also compare our proposed model with the other two different ways of setting the history influence on the generation of documents at current epoch: sliding-DTSA and skip-DTSA.

- **sliding-DTSA**: the current sentiment-topic-word distributions are dependent on the previous sentiment-topic specific word distributions in the last S epochs.

- **skip-DTSA**: we take history sentiment-topic-word distributions into account by skipping some epochs in between. For example, if S = 3, we only consider previous sentiment-topic-word distributions at epoch $t - 2^2$, $t - 2^1$ and $t - 2^0$.

Figure 5 shows the average training time per epoch with the increasing number of time slices. sliding-DTSA, skip-DTSA and DTSA have similar average training time across the number of time slices. JST-one has less training time than the DTSA models. LDA-one uses least training time since it only models 3 sentiment topics while others all model a total of 20 sentiment topics. JST-all takes much more time than all the other models as it needs to use all the previous data for training.

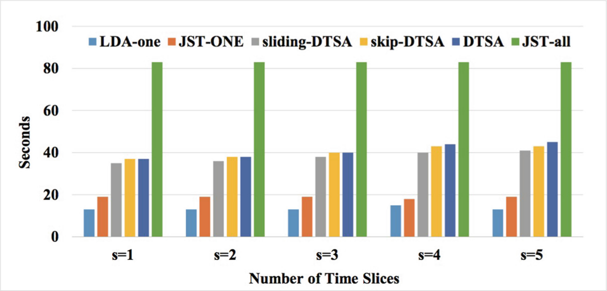


Fig. 5. Computational time per epoch with different number of time slices.

5.3 Sentiment Classification

In this section, we present the results of sentiment classification with the number of time slices fixed at S = 4. We use the above mentioned datasets (see Table 2) to do the experiment. DTSA is a probabilistic model, we run 10 times for each experiment, and list the average F1-score in Table 4.

We compare the performance of our model with JST-one and JST-all [17]. In order to prove the importance of user interactions, we introduce two special sentiments to JST-one and JST-all, making the new model called JST-one+ and JST-all+ which could identify approval and disapproval. For evaluating the advantage of using multiple timescale model, we also compare the DTSA model with sliding-DTSA and skip-DTSA.

As can be seen from Table 4, the performance of DTSA, sliding-DTSA and skip-DTSA are better than JST-one and JST-one+ method on all data sets. This is because JST-one and JST-one+ only use the data in the previous epoch for training and do not model dynamics. While our models take into account the influence of history sentiment-topic-word distributions, which can improve the sentiment classification metrics. Compared to sliding-DTSA and skip-DTSA, our model DTSA achieve the highest F1-score, which proves the effective of multiple timescale model.

Table 4. The F1-score of sentiment classification results.

Dataset	DTSA	sliding-DTSA	skip-DTSA	JST-one	JST-one+	JST-all	JST-all+
MH370/Topic	**0.865**	0.811	0.859	0.674	0.683	0.786	0.790
MH370/Senti	**0.831**	0.792	0.803	0.613	0.679	0.715	0.767
Crimea/Topic	**0.837**	0.793	0.828	0.607	0.615	0.776	0.782
Crimea/Senti	**0.812**	0.733	0.769	0.579	0.631	0.727	0.731
Sochi/Topic	**0.896**	0.795	0.869	0.683	0.690	0.765	0.765
Sochi/Senti	**0.853**	0.763	0.783	0.602	0.668	0.734	0.760
India/Topic	**0.879**	0.815	0.853	0.657	0.662	0.790	0.803
India/Senti	**0.848**	0.778	0.793	0.613	0.659	0.716	0.765
Facebook/Topic	**0.857**	0.806	0.848	0.637	0.637	0.753	0.762
Facebook/Senti	**0.828**	0.749	0.776	0.591	0.629	0.687	0.733

In addition, we can see that JST-one+ and JST-all+ significantly improve the accuracy of sentiment classification on all data sets. This suggests that the special sentiments have a significant impact to the sentiment classification result. Furthermore, the DTSA outperforms the JST-all and JST-all+ methods on all data sets. JST-all+ could detect the user interactions, but does not use the user interactions to adjust the topic and sentiment distribution of comments, making them can not avoid the error caused by user interaction on both topic and sentiment.

We also analyze the influence of the topic number settings on the DTSA model performance. With the number of time slices fixed at S = 4, we vary the topic number $T \in \{1, 5, 10, 15, 20, 25\}$. Figure 6 shows the average sentiment classification accuracy over epochs with different number of topics. As can be seen from Fig. 6, increasing the number of topics leads to a slight drop in accuracy. This trend is more evident on the twitter data set.

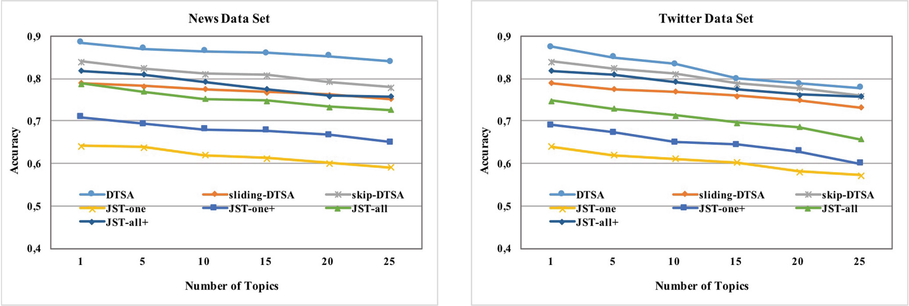


Fig. 6. Sentiment classification accuracy with different number of topics.

6 Conclusion

In this paper, a novel dynamic topic-based sentiment analysis model (DTSA) is proposed to extract topics and topic-specific sentiments from online news stories and comments. It could be used to decrease the error caused by user interactions, handle long-term and short-term dependency and automatically adjust model parameters in real time to improve the accuracy of classification based on sentiment recognition. The model is deployed on distributed online systems thus making improvements of efficiency of data process. The model has been tested on two kinds of data sets and displays promising results.

In the future, more semantic information such as relationships between terms and more features like sarcasm could be used into this model to further improve the accuracy of sentiment analysis. Besides, we should study the performance of the various evaluation methods in topic words extraction tasks.

References

1. Lin, C., He, Y.: Joint sentiment/topic model for sentiment analysis. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 375–384. ACM (2009)
2. Jo, Y., Oh, A.H.: Aspect and sentiment unification model for online review analysis. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 815–824. ACM (2011)
3. Li, C., Zhang, J., Sun, J.T., et al.: Sentiment topic model with decomposed prior. In: SIAM International Conference on Data Mining (SDM 2013). Society for Industrial and Applied Mathematics (2013)
4. Balahur, A., Steinberger, R., Kabadjov, M., et al.: Sentiment analysis in the news (2013). arXiv preprint: arXiv:1309.6202
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
6. Blei, D.M.: Probabilistic topic models. Commun. ACM **55**(4), 77–84 (2012)
7. Titov, I., McDonald, R.: Modeling online reviews with multi-grain topic models. In: Proceedings of the 17th International Conference on World Wide Web, pp. 111–120. ACM (2008)
8. Kim, S., Zhang, J., Chen, Z., et al.: A hierarchical aspect-sentiment model for online reviews. In: AAAI (2013)
9. Zhao, Y., Dong, S., Li, L.: Sentiment analysis on news comments based on supervised learning method. Int. J. Multimed. Ubiquit. Eng. **9**, 333–346 (2014)
10. Wang, C., Blei, D., Heckerman, D.: Continuous time dynamic topic models (2012). arXiv preprint: arXiv:1206.3298
11. Wang, X., McCallum, A.: Topics over time: a non-Markov continuous-time model of topical trends. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 424–433. ACM (2006)
12. Iwata, T., Yamada, T., Sakurai, Y., et al.: Online multiscale dynamic topic models. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 663–672. ACM (2010)

13. Wang, Y., Agichtein, E., Benzi, M.: TM-LDA: efficient online modeling of latent topic transitions in social media. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 123–131. ACM (2012)
14. Dermouche, M., Velcin, J., Khouas, L., et al.: A joint model for topic-sentiment evolution over time. In: 2014 IEEE International Conference on Data Mining (ICDM), pp. 773–778. IEEE (2014)
15. Zheng, M., Wu, C., Liu, Y., et al.: Topic sentiment trend model: modeling facets and sentiment dynamics. In: 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), pp. 651–657. IEEE (2012)
16. Wang, L., Cardie, C.: Improving agreement and disagreement identification in online discussions with a socially-tuned sentiment lexicon. In: ACL 2014, p. 97 (2014)
17. Lin, C., He, Y., Everson, R., et al.: Weakly supervised joint sentiment-topic detection from text. IEEE Trans. Knowl. Data Eng. **24**(6), 1134–1145 (2012)
18. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)

Improving Object and Event Monitoring on Twitter Through Lexical Analysis and User Profiling

Yihong Zhang(✉), Claudia Szabo, and Quan Z. Sheng

School of Computer Science, The University of Adelaide, Adelaide, SA 5005, Australia
{yihong.zhang,claudia.szabo,michael.sheng}@adelaide.edu.au

Abstract. Personal users on Twitter frequently post observations about their immediate environment as part of the 500 million tweets posted everyday. These observations and their implicitly associated time and location data are a valuable source of information for monitoring objects and events, such as earthquake, hailstorm, and shooting incidents. However, given the informal and uncertain expressions used in personal Twitter messages, and the various type of accounts existing on Twitter, capturing personal observations of objects and events is challenging. In contrast to the existing supervised approaches, which require significant efforts for annotating examples, in this paper, we propose an unsupervised approach for filtering personal observations. Our approach employs lexical analysis, user profiling and classification components to significantly improve filtering precision. To identify personal accounts, we define and compute a mean user profile for a dataset and employ distance metrics to evaluate the similarity of the user profiles under analysis to the mean. Our extensive experiments with real Twitter data show that our approach consistently improves filtering precision of personal observations by around 22 %.

Keywords: Twitter · Microblog content classification · User profiling

1 Introduction

Micro-blogging services such as Twitter have become widely used in recent years. Twitter allows its users to create and publish short messages of maximum 140 characters, called *tweets*. Currently, around 284 million active Twitter users generate 500 million tweets every day[1], and around 80 % of Twitter users use their mobile phones to create tweets[1]. The use of mobile platforms for tweeting implies that users can report observed events and objects in their physical vicinity. For example, personal tweets have been used for tracking the movements of earthquakes and typhoons in Japan [12]; tweets about flood, hurricane, and riots have also been used for crime and disaster location [7]. In a previous work, we

[1] https://about.twitter.com/company.

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 19–34, 2016.
DOI: 10.1007/978-3-319-48743-4_2

showed that news generated based on personal observation messages on Twitter can often be hours earlier than the first news appearing in traditional media, even for the most newsworthy events such as shooting incidents [19].

Current work on microblog and short text analysis mostly relies on supervised machine learning methods [2,12], which require the manual preparation of training samples. This has several drawbacks, such as the need for significant manual effort for annotating examples, and a lack of quality guarantees of the classification solutions, when the classifier is applied to a wider pool of tweets beyond its training data. *Unsupervised methods* have the potential to address these issues. In the domain of microblogs and short texts, however, due to the complexity and uncertainty of human user data, works on unsupervised methods are still at an early stage [1,17]. The informal and unstructured text messages used on microblogs creates uncertainty for any kind of classification models, and solutions and models effective in one application often will not be as effective in another application. For example, the work in [8] has shown that the effect of user roles in Twitter rumor classification varies significantly for different rumor instances. Thus a challenge for a specific application using unsupervised methods is to find a particular model that is effective for that application and domain.

In this paper, we focus on filtering personal observations of objects and events on Twitter using an unsupervised method. To address the challenges discussed above and provide high classification accuracy, we advance a novel approach that employs lexical analysis and user profiling. The lexical analysis module filters *observation messages* based on two attributes, part-of-speech (POS) tag and message objectivity. The user profiling module separates *personal accounts* from other types of accounts based on four analyzed attributes, namely, *objectivity*, *interactivity*, *originality*, and *topic focus*. We conduct extensive experiments using real Twitter data collected for a variety of events, with significantly improved results over the existing works. Our main contributions are:

- We propose a novel unsupervised method for filtering personal observations on Twitter. Our method utilizes various natural language processing techniques in lexical analysis and user profiling.
- We propose a novel model for profiling Twitter users based on four dimensions, including *objectivity*, *interactivity*, *originality*, and *topic focus*. We also propose algorithms that can effectively distinguish personal accounts from specific-purpose accounts.
- We test our method extensively with real Twitter datasets. For controlled datasets, our method consistently improves the precision by around 22 %. We obtain even higher improvement for crowd-sourced datasets. Our method also out-performs some of the most effective supervised techniques.

2 Related Work

Twitter as a public media and news source has been studied in several works. Wu et al. [18] investigated the demographics of influential Twitter users, whom they grouped into media, organization, celebrity, and blogger. Their study concludes that bloggers are popular personal accounts that produce the most influential tweets. Kwon et al. [5] investigated supervised methods for identifying rumors on Twitter using a number of prominent features, including propagation peaks, friendship network graph, and linguistic properties based on LIWC (Linguistic Inquiry and Word Count). They found that selecting the right features is critical to classifier accuracy. Sriram et al. [14] similarly studied supervised tweet classification for five types of tweets: news, opinions, deals, events, and private messages. Although various machine learning techniques are compared, they also found selecting the right features is the key factor for classification accuracy.

Despite the fact that messages on Twitter are very often informal and incomplete [3], researches have used Twitter information for disaster location [6], object tracking [12], and event detection [16]. For example, Lingad et al. [7] studied locations mentioned in disaster-related messages in order to identify the position of natural disasters and affected areas. However, accurately classifying object or event-related messages is a challenging task. Sakaki et al. [12] developed a system that tracks the movement of earthquakes and typhoons based on personal reports detected on Twitter. They compared a number of features for building the event-related message classifier, with the best-performing feature set achieving a precision of 63.64 %. Li et al. [6] studied the use of tweets for detecting crime and disaster events (CDE) as they were reported on Twitter. They trained a classifier based on the words present in identified CDE tweets, and achieved a precision of 30 % and a recall of 85 %.

Due to the amount of effort required for manually annotating a large number of messages, a supervised method, however, is in many cases impractical. Unsupervised methods have the advantage of needing less manual effort. Current unsupervised methods for microblog analysis are still at an early stage of research. Carroll et al. [1] developed an unsupervised method for determining the objectivity of in Chinese microblog texts. They defined objectivity as sentiment neutrality, but the application is limited to brand and company reputation analysis. Unankard et al. [16] developed a framework for predicting election results using Twitter messages, in which message and user sentiments are calculated based on positive and negative word counts. Since observation messages are not strongly related to message sentiments, filtering personal observations, however, requires technique beyond sentiment analysis.

3 Filtering Methodology

Our method filters observations of objects and events from personal accounts, by performing the following steps. First, we identify observations from collected tweets for a specific keyword. Second, using also the collected tweets, we distinguish personal accounts from other types of accounts. A personal account is a Twitter account employed for personal use, and is assumed to be free from business or propaganda interests. Our insight is that tweets from personal accounts often contain realtime and localized observations of objects and events. Finally, from the observation tweets identified in the first step, we retain only those made from personal accounts. These personal observations of objects and events have proved useful in previous works for scenarios such as disaster location and rumor detection [5,12].

An overview of our method is shown in Fig. 1. To identify observation tweets, we run lexical analysis on tweet texts based on the par-of-speech (POS) tagging, objectivity analysis, and originality test. To identify personal accounts, we first analyze four attributes for each user, namely, *objectivity*, *interactivity*, *originality*, and *topic focus*. Then we use a clustering algorithm for classifying personal accounts based on the attribute values. We describe our method below.

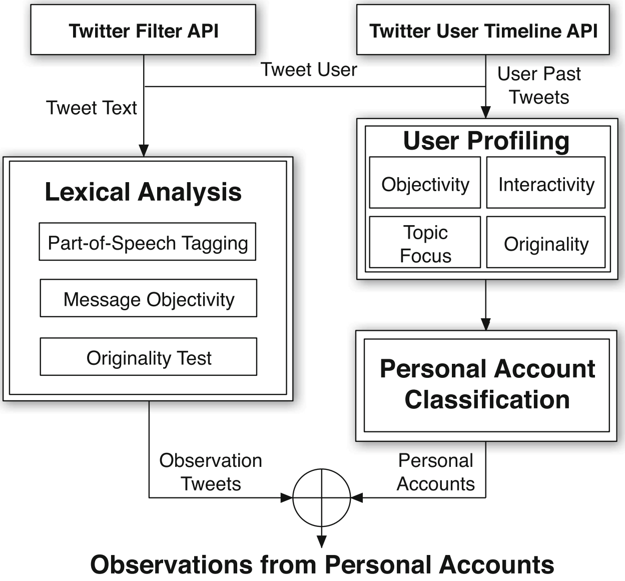


Fig. 1. Method overview

3.1 Observation Filtering

After using the Twitter Filter API[2] to obtain tweets that contain the object or event keyword such as "rainbow" or "car accident", our lexical analysis method focuses on extracting observation tweets. Not all tweets containing the keywords are observations of objects and events, since in some cases the keywords can have another semantic, context-based meaning, and the objects and events can be mentioned in general comments instead of specific observations, e.g., "I dislike car accidents". We address this by utilizing three techniques, namely, *par-of-speech* (POS) tagging, *objectivity analysis*, and *originality test*. POS tagging allows filtering of messages in which the object or event keyword is not used as a subject of observation. Objectivity analysis allows filtering of uncertain messages, such as questions and general comments. Originality test removes messages that are not originally created by the user, such as retweets or quotations.

[2] https://dev.twitter.com/streaming/public.

Filtering Based on Part-of-Speech Tagging. Our insight is that the objects and events mentioned in an observation are most likely to be nouns and gerunds, such as in "I just saw a rainbow", or "A shooting outside my home". On the other hand, keywords not used as nouns and gerunds often indicate that the tweet is not a specific observation. Some examples of non-observation tweets are shown in Table 1, with the role of the keyword determined by POS tagging.

Table 1. Non-observation tweets filtered by POS tagging, for monitoring flight delay, shooting incidents, and rainbows

Tweet text	POS
Keep praying for the typhoon to magically **delay** my flight a day	VB
Can we pretend that airplanes, in the night sky, are like **shooting** stars?	JJ
This guy got on a **rainbow** colored LV belt	JJ

VB = base form verb, JJ = adjective.

POS tagging is a technique that matches words in a text with their part-of-speech categories, such as modal, noun, verb, and adverb [13]. We use a filtering rule on top of POS tagging to effectively remove a portion of tweets that are clearly not observations. After performing POS tagging for a tweet, we accept it if the POS tag for the keyword is **NN** (Noun, singular or mass), **NNP** (proper noun, singular), and **VBG** (verb, gerund or present particle). The tweet is rejected if the keyword has other POS tags.

Filtering Based on Objectivity Analysis. Our insight is that a specific observation of an object or event usually is written in a more objective tone than a general tweet. Generally, the objectivity of a message is affected by sentimental words and uncertain words, such as "great", "bad", "maybe", "anyone". Sentimentality and uncertainty as factors for determining message objectivity has already been proposed in existing works [1,10]. We calculate tweet objectivity based on both sentimentality and uncertainty, using the following formula:

$$\begin{aligned} objectivity(t) = 1 - [senti_p(t) + 0.5 \times senti_n(t)] \\ \times (1 - \sqrt{uncertainty(t)}) \end{aligned}$$

where $senti_p$ is the positive sentiment and $senti_n$ is the negative sentiment. In our previous works, we have found that negative sentiments have a large presence in observation messages [20]. We follow this insight here and we weight down the effect of negative sentiments on reducing the objectivity in the formula. Furthermore, since uncertainty plays an important role in determining the objectivity of a message, as discovered in [10], we increase the effect of uncertainty by scaling it to a larger value.

For sentiment analysis, we employ previously proven effective methods, which employ a positive/negative words dictionary and the slang sentiment dictionary [16]. The positive and negative sentiments of a tweet text t are measured as:

$$senti_p(t) = \frac{count_p(t)}{count_w(t)}$$

$$senti_n(t) = \frac{count_n(t)}{count_w(t)}$$

where $count_p(t)$ and $count_n(t)$ are the word count for positive and negative words in t, and $count_w(t)$ is the word count of t.

For uncertainty analysis, we use a dictionary of uncertain words based on the LIWC category of hesitation words [15]. To measure the uncertainty of tweet t, we consider the number of uncertain words in the text, and whether it is a question.

$$uncertainty(t) = \begin{cases} 0.5, \text{ if } t \text{ ends with a question mark} \\ \frac{count_u(t)}{count_w(t)}, \text{ otherwise} \end{cases}$$

where $count_w(t)$ is the word count for uncertain words in t.

Originality Test. Our analysis of various datasets show that sometimes personal users may repeat some messages created by other users, which do not count as their own observations. The repeated messages not only produce redundancy, but also generate noises for analysis. Thus it is crucial to determine message originality. We proposed a set of rules to determine non-original messages based on message content, as shown in Table 2. A message satisfies any of the rules in the table is considered non-original, and will be filtered out.

Table 2. Originality test rules

Rule	Explanation
Retweet	Contains the word RT
Quotation	Contains quotation marks
Speech	Mention or capitalized word before colon
News title	All words capitalized before link
Repeat	Contains "says", "claims", "via", or "according to"
News mention	Mention contains "news", "radio", or "breaking"
News agent	Mention contains news agent name such as "ABC" or "CNN"

Some repeated messages are easy to identify, such as retweets, which have "RT" at the beginning of the messages. Other forms of repeated messages can be more difficult to spot, such as indirect quotes, which often but not necessarily contain the word "says" or "claims". Given the various ways a message may be repeated, the rules listed in Table 2 do not cover all non-original messages. Nevertheless, we found these rules to filter out most of the repeated messages.

Algorithm 1. Lexical Analysis on Single Tweets

```
INPUT: keyword $w$, tweet set $T$, objectivity threshold $\theta$
OUTPUT: obervation labels $O$
 1: set all $o \in O$ as $false$
 2: for each $t \in T$ do
 3:     run POS tagging for $t$
 4:     if POS tag for $w \in \{\mathbf{NN}, \mathbf{NNP}, \mathbf{VBG}\}$ then
 5:         $pp \leftarrow true$
 6:     end if
 7:     if $objectivity(t) > \theta$ then
 8:         $po \leftarrow true$
 9:     end if
10:     if $t$ fails all rules in Table 2 then
11:         $pt \leftarrow true$
12:     end if
13:     $o_t \leftarrow pp \wedge po \wedge pt$
14: end for
```

Lexical Analysis Algorithm. Algorithm 1 describes our lexical analysis method. The input is a keyword w, and a set T of tweet texts containing the keyword. The output is a set of predictions of whether each tweet text $t \in T$ is an observation, O. In line 7, we use a parameter θ to control the level of objectivity a tweet requires to meet to be considered an observation. The default value for θ is the first quartile of overall objectivity in the tweet set.

3.2 User Profiling for Personal Account Classification

Previous works have shown that news generated from personal observations on Twitter can be much faster than traditional media, and the implicitly-associated location data can be used for localizing the object or the event [12,19]. However, there are many Twitter accounts that are not for personal use, and do not have the same time and location association for their observation messages, and while they add noises to the data collected, it is usually difficult to distinguish them from personal accounts. The main issue is that all accounts on Twitter uses the same format to store data, and usually there is no effective way to judge the type of account other than looking at the content of the account posts directly. These accounts include news, business, activist and advertisement accounts. We call these latter types of accounts *specific-purpose accounts*, and show some well-known examples in Table 3.

Table 3. Examples of specific-purpose accounts

News	@cnnbrk @wsj @foxnews @huffingtonpost @bbcworld @politico
Business	@AdamDenison @GMblogs @MarriottIntl @chicagobulls @Marvel
Activist	@Greenpeace @femmajority @OU_Unheard @freedomtomarry

Our study of personal and specific-purpose accounts leads to the following observations:

– News accounts tweet about various topics in a strictly objective tone. Their tweets usually contain links to Web articles. Depending on the specialty, a media account can cover a wide range of topics.
– Business accounts contain conversations, observations, and product promotions, but the range of topic is limited to the specific business.
– Activist and advertisement accounts rarely use objective tone, and their range of topics is also limited.

A personal account, however, does not have such clear-cut characteristics as specific-purpose accounts, and usually contains a mix of information sharing, conversation with other users, and original content that covers various topics. We propose that:

Conjecture 1. A personal account has moderate levels in objectivity, interactivity, originality, and topic focus.

We use various statistics generated from Twitter data to calculate the levels of *objectivity*, *interactivity*, *originality*, and *topic focus* for Twitter users. Here we assume these user qualities are consistent over time and do not easily change. There are rare cases that the profile of a user changes drastically, for example, caused by a job change, but currently we do not consider such cases. To profile a user, first we collect a set of past tweets made by the user, H. Then we select the original tweets in H based on the rules described in Table 2, as $OH = \{oh_1, oh_2, ..., oh_l\}$, where $|OH| = l$.

The objectivity of a user is calculated based on the objectivity of each tweet in OH:

$$u_{objectivity} = \frac{\sum_{i=1}^{l} objectivity(oh_i)}{l}$$

The interactivity of a user is calculated based on the number of tweets containing mention mark "@" in H:

$$u_{interactivity} = \frac{count_{@}(H)}{|H|}$$

The originality of a user is calculated based on the fraction of original tweets in H.

$$u_{originality} = \frac{l}{|H|}$$

To calculate a user's topic focus, we count the frequency of each topic word for all topic words appearing in OH. For simplicity, we consider a topic word as a word that starts with a capital letter. The first word in a sentence is ignored. Once we have a descendingly-sorted list of topic word occurrences $\{nt_1, nt_2, ..., nt_k\}$, the topic focus of a user is calculated based on the fraction of the first quarter of the most frequent topic words:

$$u_{focus} = \frac{\sum_{i=1}^{n/4} nt_i}{\sum_{j=1}^{n} nt_j}$$

A user is thus profiled by the quadruple:

$$u = \{u_{objectivity}, u_{interactivity}, u_{originality}, u_{focus}\}$$

3.3 Personal Account Classification with Profiles

We propose an algorithm for automatically identifying personal accounts based on the user profile. First we define the difference between two user profiles u_1 and u_2 as the Euclidian distance between two profiles:

$$d(u_1, u_2) = \sqrt{\sum (u_1 - u_2)^2}$$

where

$$\begin{aligned} \sum (u_1 - u_2)^2 = {} & (u_{objectivity1} - u_{objectivity2})^2 \\ & +(u_{interactivity1} - u_{interactivity2})^2 \\ & +(u_{originality1} - u_{originality2})^2 \\ & +(u_{focus1} - u_{focus2})^2 \end{aligned}$$

Following Conjecture 1, we see that the attributes of a personal account are usually closer to a set of mean values while a specific-purpose account usually holds more extreme values. Therefore we propose that:

Conjecture 2. Given a set of user profiles U, which contains personal account profiles P and specific-purpose account profiles S, there exists a mean profile $\bar{u}$, such that $\sum_{p \in P} d(p, \bar{u}) < \sum_{s \in S} d(s, \bar{u})$.

While it is difficult to prove Conjecture 2, we find it generally true in our analysis, as we will show with our experiments. Given a set of user profiles U, and a mean profile $\bar{u}$, we can separate from U a subset C that is more likely to contain personal accounts, by selecting profiles that have shorter distance to $\bar{u}$.

We devise an iterative algorithm for finding the mean profile $\bar{u}$. Intuitively, we can use the mean attribute values of all profiles in U. However, the extreme attribute values of the specific-purpose account profiles can bias the mean significantly, making it inaccurate for deciding personal accounts. In Algorithm 2, we use an iterative approach and a cluster size threshold δ for selecting a cluster of $|U| \times \delta$ profiles that are close to an unbiased $\bar{u}$. Starting from an initial mean profile $\bar{u}_0$, the algorithm alters between cluster updating (line 2, 6) and mean updating (line 4 and 5). In the cluster updating step, a number of profiles close to the mean are selected. In the mean updating step, a new mean is calculated based on the selected profiles. If there are extreme values that cause a bias in the cluster, the mean will move away from the bias, and replace the extreme value profiles with more average profiles in the cluster. The output of the algorithm, F, is a set of personal account predictions.

Algorithm 2. Predicting Personal Accounts

```
INPUT: user profiles $U$, mean profile $\bar{u}_0$, selected cluster size $\delta$
OUTPUT: $F$
set all $f \in F$ as $false$
$C \leftarrow |U| \times \delta$ profiles closest to $\bar{u}_0$
while $C \neq C'$ do
    $C' \leftarrow C$
    $\bar{u} \leftarrow$ mean attribute values of profiles in $C$
    $C \leftarrow |U| \times \delta$ profiles closest to $\bar{u}$
end while
for each $u \in U$ do
    if $u \in C$ then
        $f_u \leftarrow true$
    end if
end for
```

While Algorithm 2 generally finds a good mean profile that separates personal accounts and specific-purpose accounts. However, depending on the choice of the initial mean $\bar{u}_0$, the algorithm sometimes produces undesirable results. To address this issue, we derive a particle swarm optimization (PSO) algorithm for finding the optimal $\bar{u}_0$.

PSO is an optimization technique that takes a population of solutions, and iteratively improves the quality of the solutions by moving them toward the best solution in each iteration [4]. A solution in our PSO algorithm is an initial mean $\bar{u}_0$ to be given to Algorithm 2. A PSO algorithm requires the definition of the quality measure and the solution movement. To define the quality of a solution, we rely on our initial observation that personal accounts exhibit higher variance than any types of specific-purpose accounts. Therefore we propose that:

Conjecture 3. Given two user profile clusters C_1 and C_2, if the profiles in C_1 are more diverse than C_2, than C_1 is more likely to contain personal accounts.

We use pairwise profile differences to calculate the diversity of profiles in a cluster, $C = \{c_1, c_2, ..., c_k\}$,

$$div(C) = \frac{2 \times \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(c_i, c_j)}{k \cdot (k-1)}$$

For the solution movement in PSO, we set a moving speed v so in each iteration, a solution p moves towards the best solution p_b as:

$$p \leftarrow p + (p_b - p) \cdot v \tag{1}$$

Our PSO algorithm is shown as Algorithm 3. It starts with a number of random solutions (line 1) and for each solution, a profile cluster is generated using Algorithm 2 (line 2 to 4). Then iteratively, the PSO algorithm moves the best solution towards an optimal solution by comparing the cluster diversity with each solution (line 5 to 13).

Algorithm 3. PSO for Finding Optimal $\bar{u}_0$

```
INPUT: user profiles $U$, selected cluster size $\delta$, number of particles $n$, speed $v$
OUTPUT: $p_b$
 1: randomly choose $n$ solutions $P$ in the solution space
 2: for each $p \in P$ do
 3:     generate a cluster $C_p$ using Algorithm 2
 4: end for
 5: $p_b \leftarrow p$ with highest $div(C_p)$
 6: while $p_b \neq p'_b$ do
 7:     $p'_b \leftarrow p_b$
 8:     for each $p \in P$ do
 9:         $p \leftarrow p + (p_b - p) \cdot v$
10:         generate a cluster $C_p$ using Algorithm 2
11:     end for
12:     $p_b \leftarrow p$ with highest $div(C_p)$
13: end while
```

The optimal initial mean produced by Algorithm 3 can then be used in Algorithm 2 for selecting the cluster of personal account profiles. Although Algorithm 3 requires two more parameters, during our experiments we find the effect of changing n and v negligible for any $n > 1,000$ and $v < 0.2$, as the solution already reaches optimal values. Therefore we can confidently set n and v to fixed values. The only parameter that still affects the classification result is the cluster size parameter δ, which controls the portion of profiles in the data to be selected as personal account profiles.

Algorithm 4. Filter Observations from Personal Accounts

```
INPUT: keyword $w$, messages $M$, objectivity threshold $\theta$, selected cluster size $\delta$
OUTPUT: $R$
set all $r \in R$ as $false$
$T \leftarrow$ tweet text from $M$
$U \leftarrow$ user profiles from $M$
$O \leftarrow$ run Algorithm 1 with $w$, $T$, $\theta$
$p_b \leftarrow$ run Algorithm 3 with $U$, $\delta$
$F \leftarrow$ run Algorithm 2 with $U$, $p_b$, $\delta$
for each $m \in M$ that has text $t$ and user profile $u$ do
    if $o_t \wedge f_u$ then
        $r_m \leftarrow true$
    end if
end for
```

3.4 Overall Algorithm

Algorithm 4 identifies observations from personal accounts. Given the input of a keyword w and a set of tweets M, and the control parameter θ and δ, the output is a set of predictions, R, of whether each respective tweet is an observation of the object or event of interest from personal accounts.

4 Experimental Analysis

We tested the effectiveness of our method for filtering personal observations on Twitter with two real Twitter datasets, comprising of a controlled dataset and a crowd-sourced dataset. In this section, we present the setup, measurement, baseline methods, and results of our experiments in detail.

4.1 Experiment Setup

We implemented the algorithms presented in the previous section in Java. The experiments were run on a MacBook Pro laptop computer, with 2.3 GHz Intel Core i7 CPU and 8 GB 1600 MHz DDR3 memory. We deployed an existing implementation for POS tagging. After comparing several existing POS tagging implementations including OpenNLP and LingPipe, we chose StanfordNLP POS module to run our POS tagging because it is relatively fast and provides a high tagging accuracy of around 95 % [9].

For parameter θ in Algorithm 1, we chose the first quartile of overall objectivity in the dataset for all experiments, which generally provides good results. For parameter δ, we compared three different values, including 0.7, 0.8, and 0.9. To ensure the consistency of the experiments, instead of randomly choosing initial values for the particles in Algorithm 3, we chose combinations of evenly distributed values for the four attributes as the initial values, i.e., 0.2, 0.4, 0.6, 0.8, and 1. Our analysis shows that randomly initialized particles provide similar results. For user profiling, up to 1,000 recent tweets were collected for each user using Twitter Timeline API.

4.2 Baseline Methods and Comparison Metrics

We compared our approach with three baseline filtering strategies, namely, *Accept All*, *Sakaki* filter, and *Sriram*. Accept All takes all tweets in the dataset as the positive for personal observations. Sakaki classifier, proposed by Sakaki et al. in [12], is a supervised method that deploys a Support Vector Machine (SVM) classifier with linear kernel built on manually annotated training data. Among the three feature sets proposed in [12], we implemented the reportedly most effective set, Feature Set A, which is based on word counts and keyword positions. We deployed an existing SVM implementation in an R language package called e1071[3]. We used a weighting function according to class imbalance to ensure optimal performance of the classifier. The performance of the Sakaki classifier was measured using the three-fold cross validation. One drawback of the

[3] https://cran.r-project.org/package=e1071.

Sakaki classifier is that it requires the presence of a keyword. The user profiling in our approach, though, does not have this requirement.

The Sriram classifier, proposed by Sriram et al. in [14], is also a supervised method that is based on eight features and the Naive Bayes model. The eight features include author name, use of slang, time phrase, opinionated words, and word emphasis, presences of currency signs, percentage signs, mention sign at the beginning and the middle of the message. The evaluation is based on the five-fold cross validation. The Sriram classifier is shown to be effective in classifying tweets into categories such as news, opinions, deals and events, but has not been tested in other applications.

All datasets for evaluation were manually annotated according to whether each tweet is a personal observation of an object or event of interest, which were considered ground truth in our experiments. The output of the filtering methods were compared with the manual annotations. If a filtering output is positive in manual annotations, it is considered a *true positive*. We use *precision*, *recall* and $f - value$ as the measurements of filtering accuracy, where given the set of positive filtering results P and the set of true positives in the dataset TP, The $precision = \frac{|P \bigcap TP|}{|P|}$, $recall = \frac{|P \bigcap TP|}{|TP|}$, and f-value $= 2 \cdot \frac{precision \cdot recall}{precision + recall}$.

4.3 Effectiveness on Controlled Datasets

We first tested our method on two controlled datasets. We collected a dataset of around 5,000 tweets containing the keyword *hailstorm* during August, 2015, and a dataset of around 5,000 tweets containing the keyword *car accident* during September, 2015. After removing retweets and tweets containing links, we manually labelled the remaining tweets as positive or negative examples, according to whether the message is about a direct observation of a hailstorm or a car accident. The resulted *hailstorm* dataset contains 675 tweets, with 251 positive examples and 424 negative examples. The labelled *accident* dataset contains 954 tweets, with 347 positive examples and 607 negative examples.

We tested the filtering methods on the two datasets. Accuracy results for the baseline methods, lexical analysis-only filtering (LX), and lexical analysis combined with personal account filtering using three δ values, PA($\delta = 0.9$), PA($\delta = 0.8$), and PA($\delta = 0.5$), are presented in Table 4.

As shown in the table, the Accept All strategy captured all the positives in the annotations and had the maximum recall of 1. All other methods improved the precision by sacrificing the recall to some degree. Personal account filtering with δ set to 0.9 achieved the highest overall performance, indicated by the highest f-value. Using lexical analysis only and PA with $\delta = 0.9$ and $delta = 0.8$ all performed better than the Sakaki classifier and the Sriram classifier, the latter of which provided almost no filtering effect in the hailstorm dataset. Setting δ to a lower value improved the precision but also lowered the recall. When setting δ to 0.5, PA achieved the highest precision, while still held a relatively high f-value. The performance of all methods were consistent across two datasets, with LX improving precision from the Accept All strategy by around 15 % and PA($\delta = 0.9$) further improved it by 5 %–7 %.

Table 4. Filtering accuracy for hailstorm and car accident datasets

	Accept all	Sakaki	Sriram	LX	PA($\delta = 0.9$)	PA($\delta = 0.8$)	PA($\delta = 0.5$)
`Hailstorm dataset`							
Precision	0.37	0.43	0.37	0.53	0.62	0.64	**0.70**
Recall	**1**	0.70	0.98	0.80	0.76	0.71	0.46
f-value	0.54	0.53	0.54	0.63	**0.68**	0.67	0.56
`Car accident dataset`							
Precision	0.38	0.50	0.44	0.53	0.58	0.59	**0.60**
Recall	**1**	0.73	0.84	0.76	0.74	0.69	0.43
f-value	0.55	0.60	0.57	0.63	**0.65**	0.64	0.50

4.4 Effectiveness on Crowd-Sourced Dataset

We tested our approach on a publicly available dataset produced by Castillo et al. [11], and is available online[4]. The dataset contains around 20,000 tweets related to crisis events, such as the Colorado wildfires and the Pablo typhoon in 2012, and the Australia bushfire and New York train crash in 2013. These crisis tweets were manually annotated by hired workers on Crowdflower, a crowd-sourcing platform[5]. The tweets were labelled according to their relevance to the crisis event, and the type of information they provide into four categories, namely, *related and informative*, *related but not informative*, *not related*, and *not applicable*. The seven information types include Eyewitness, Government, NGOs, Business, Media, Outsiders, and Not applicable.

We consider that the Eyewitness-type tweets in the dataset are personal observations, while other types of tweets are not. Hence we expect our approach to filter Eyewitness tweets from other tweets. With this goal, we re-organized the dataset. First we selected two categories of related tweets from the dataset. Then we selected five information types of tweets: Eyewitness, Government, NGOs, Business and Media. We then produced a list of labels, with Eyewitness tweets as positives, and other types of tweets negatives. We also removed retweets from the data. Our labelled dataset had 3,646 tweets with 528 positives.

Since the tweets do not contain a specific keyword, we did not run POS and objectivity analysis. The Sakaki classifier is not applicable without a keyword. As such we ran the originality test in the lexical analysis and the personal account classification, and compared only to the Sriram classifier (Table 5).

The results are similar to previous experiments, where PA($\delta = 0.9$) achieved the highest f-value and PA($\delta = 0.5$) achieved the highest precision. The lexical analysis was particularly effective for this dataset, improving the precision by around 38 %, mainly because the dataset includes a large portion of news messages, which failed the originality test. After the lexical analysis, PA($\delta = 0.9$) further improved the precision by 12 %. Both LX and PA($\delta = 0.9$) significantly outperformed the Sriram classifier.

[4] http://crisislex.org/.

[5] http://www.crowdflower.com/.

Table 5. Filtering accuracy for the crisis dataset

	Accept all	Sriram	LX	PA($\delta = 0.9$)	PA($\delta = 0.8$)	PA($\delta = 0.5$)
Precision	0.14	0.32	0.52	0.64	0.64	**0.65**
Recall	**1**	0.52	0.47	0.50	0.48	0.27
f-value	0.24	0.40	0.47	**0.56**	0.55	0.38

5 Conclusion

Personal observations of objects and events published on micro-blogging platforms such as Twitter are an invaluable information source, and can be utilized in applications such as natural disaster tracking and crime monitoring. However, given the various ways users post messages and the large variety of account types, information about a particular object or event is usually noisy and misleading. Thus it is critical to develop a novel approach that filters out noises before the information can be further utilized. Current filtering approaches based on supervised machine learning techniques require large manual efforts and thus are impractical in many scenarios.

In this paper, we propose an unsupervised message filtering approach that consists of a lexical analysis module, which examines the message, and a personal account classification module, which examines the message history of the user and determines if the user account is a personal account. We tested our approach extensively on real Twitter datasets. For the controlled dataset, our method consistently improves the precision by around 22 %, with the lexical analysis module improves it by 15 %, and personal account classification further improves it by 7 %. We see even higher improvement in a crowd-sourced dataset, increasing the precision from 14 % to 65 %. Compared with the Sakaki classifier and the Sriram classifier, our approach was able to achieve more than 10 % higher accuracy. In the future, we will continue to investigate unsupervised methods for further filtering accuracy improvement by incorporating location and name-entity analysis.

References

1. Carroll, T.Z.J.: Unsupervised classification of sentiment and objectivity in Chinese text. In: Third International Joint Conference on Natural Language Processing, p. 304 (2008)
2. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on Twitter. In: Proceedings of the 20th International World Wide Web Conference, pp. 675–684 (2011)
3. Chung, D.S., Nah, S.: Media credibility and journalistic role conceptions: views on citizen and professional journalists among citizen contributors. J. Mass Media Ethics **28**(4), 271–288 (2013)
4. Kennedy, J.: Particle swarm optimization. In: Sammut, C., Webb, G.I. (eds.) Encyclopedia of Machine Learning, pp. 760–766. Springer, Heidelberg (2010)

5. Kwon, S., Cha, M., Jung, K., Chen, W., Wang, Y.: Prominent features of rumor propagation in online social media. In: Proceedings of 13th International Conference on Data Mining, pp. 1103–1108 (2013)
6. Li, R., Lei, K.H., Khadiwala, R., Chang, K.-C.: TEDAS: a Twitter-based event detection and analysis system. In: Proceedings of 28th International Conference on Data Engineering, pp. 1273–1276 (2012)
7. Lingad, J., Karimi, S., Yin, J.: Location extraction from disaster-related microblogs. In: Proceedings of the 22nd International World Wide Web Conference Companion, pp. 1017–1020 (2013)
8. Maddock, J., Starbird, K., Al-Hassani, H., Sandoval, D.E., Orand, M., Mason, R.M.: Characterizing online rumoring behavior using multi-dimensional signatures. In: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing, pp. 228–241 (2015)
9. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55–60 (2014)
10. Mukherjee, S., Weikum, G., Danescu-Niculescu-Mizil, C.: People on drugs: credibility of user statements in health communities. In: Proceedings of the 20th ACM International Conference on Knowledge Discovery and Data Mining, pp. 65–74 (2014)
11. Olteanu, A., Castillo, C., Diaz, F., Vieweg, S.: CrisisLex: a lexicon for collecting and filtering microblogged communications in crises. In: Proceedings of the 8th International AAAI Conference on Weblogs and Social Media, pp. 376–385 (2014)
12. Sakaki, T., Okazaki, M., Matsuo, Y.: Tweet analysis for real-time event detection and earthquake reporting system development. IEEE Trans. Knowl. Data Eng. **25**(4), 919–931 (2013)
13. Santorini, B.: Part-of-speech tagging guidelines for the penn treebank project (3rd revision). Technical report MS-CIS-90-47, University of Pennsylvania Department of Computer and Information Science Technical (1990)
14. Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., Demirbas, M.: Short text classification in Twitter to improve information filtering. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 841–842 (2010)
15. Tausczik, Y.R., Pennebaker, J.W.: The psychological meaning of words: LIWC and computerized text analysis methods. J. Lang. Soc. Psychol. **29**(1), 24–54 (2010)
16. Unankard, S., Li, X., Sharaf, M., Zhong, J., Li, X.: Predicting elections from social networks based on sub-event detection and sentiment analysis. In: Benatallah, B., Bestavros, A., Manolopoulos, Y., Vakali, A., Zhang, Y. (eds.) WISE 2014. LNCS, vol. 8787, pp. 1–16. Springer, Heidelberg (2014). doi:10.1007/978-3-319-11746-1_1
17. Unankard, S., Li, X., Sharaf, M.A.: Emerging event detection in social networks with location sensitivity. World Wide Web Journal (2015, in press)
18. Wu, S., Hofman, J.M., Mason, W.A., Watts, D.J.: Who says what to whom on Twitter. In: Proceedings of the 20th International World Wide Web Conference, pp. 705–714 (2011)
19. Zhang, Y., Szabo, C., Sheng, Q.Z.: Sense and focus: towards effective location inference and event detection on Twitter. In: The Proceedings of the 16th International Conference on Web Information Systems Engineering (2015)
20. Zhang, Y., Szabo, C., Sheng, Q.Z., Fang, X.S.: Classifying perspectives on Twitter: immediate observation, affection, and speculation. In: The Proceedings of the 16th International Conference on Web Information Systems Engineering (2015)

Aspect-Based Sentiment Analysis Using Lexico-Semantic Patterns

Kim Schouten(✉), Frederique Baas, Olivier Bus, Alexander Osinga,
Nikki van de Ven, Steffie van Loenhout, Lisanne Vrolijk, and Flavius Frasincar

Erasmus University Rotterdam, PO Box 1738, 3000 DR Rotterdam, The Netherlands
{414224fb,419325ob,427832ao,356202nv,431068sl,413096lv}@student.eur.nl,
{schouten,frasincar}@ese.eur.nl

Abstract. With its ever growing amount of user-generated content, the Web has become a trove of consumer information. The free text format in which most of this content is written, however, prevents straightforward analysis. Instead, natural language processing techniques are required to quantify the textual information embedded within text. This research focuses on extracting the sentiment that can be found in consumer reviews. In particular, we focus on finding the sentiment associated with the various aspects of the product or service a consumer writes about. Using a standard Support Vector Machine for classification, we propose six different types of patterns: lexical, syntactical, synset, sentiment, hybrid, surface. We demonstrate that several of these lexico-syntactic patterns can be used to improve sentiment classification for aspects.

Keywords: Lexico-semantic patterns · Support Vector Machines · Aspect-based sentiment analysis

1 Introduction

With its ever growing amount of user-generated content, the Web has become a trove of consumer information. Consumers everywhere are invited to share their experiences with products or services they bought and these experiences are in turn shared with prospective buyers to inform their decision making. In this, the Web has transformed the marketplace, putting the electronic word-of-mouth at the core of the decision making process. While reviews are marketed as being useful for prospective consumers, companies are even more interested in all of the expressed opinions toward their products and services. That information enables them to improve their products and optimize marketing strategies.

Unfortunately, the free text format of reviews prevents direct analysis of sentiment. Hence, data mining and natural language processing techniques are used to extract the highly valuable sentiment information. Before sentiment can be extracted, however, the sentiment scope has to be determined, since sentiment can be extracted for complete documents, sentences, or aspects. The advantages

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 35–42, 2016.
DOI: 10.1007/978-3-319-48743-4_3

of the first two options are that they are easier to do. The disadvantage is that they can not cope with situations where within the unit of analysis (i.e., the document or the sentence), two or more things are discussed that have conflicting sentiment values. To deal with this, sentiment analysis has moved to the aspect level, where sentiment is associated with actual characteristics of the product or service under review. This naturally solves the problem of conflicting sentiment, but the process becomes more complex since the aspects themselves have to be found first. In our research, we focus on the sentiment analysis only, using the aspects that are already provided in the labeled data.

More specifically, we want to investigate the use of lexico-semantic patterns for sentiment analysis, based on the hypothesis that people tend to use similar linguistic structures to express sentiment. For this, we look at lexical patterns, Part-of-Speech (i.e., word types like nouns, verbs, etc.) patterns, and synset (i.e., a set of synonyms that have a single meaning) patterns, and, in addition, at combinations of these. For example, a pattern like ‘low’ followed by ‘quality’ denotes a different sentiment than ‘low’ followed by ‘price’. This shows the difficulty of sentiment analysis and it forms the basis why we want to consider various combinations of attributes. We pose that an extended analysis of patterns will contribute to the existing sentiment analysis literature.

The paper is structured as follows. We start by discussing some of the related work in Sect. 2, followed by the description of the types of features we want to investigate in Sect. 3. We then describe our methodology and its evaluation in Sect. 4. We give our conclusions and possible directions for future work in Sect. 5.

2 Related Work

This work is a continuation of [7], which argues that patterns, either over adjacent words or over the grammatical structure of a text, can be employed together with a classifier to perform sentiment analysis. The scope in that work is still the sentence level, with all the advantages and disadvantages as discussed in the previous section. The features used are synset-based features, lexical features, and features that use the grammatical structure instead of word adjacency. We extend this research by first moving to the aspect level. Furthermore, we investigate n-grams up to $n = 4$, including some hybrid patterns like a synset followed by a Part-of-Speech tag. However, we only use word adjacency for our patterns, so grammatical relations are not employed to create patterns of non-adjacent words.

Using n-grams instead of just unigrams has been shown to increase performance and it is straightforward to implement [2,4]. For example, [4] uses both unigrams and bigrams to estimate aspect sentiment. However, the unigram feature still proved to be the most important in the ablation experiment, where this feature was left out to measure the drop in performance compared to including it in the feature set.

Part-of-Speech information, or grammatical word categories, has been used in text classification for a long time. In [3], for example, Part-of-Speech is used to filter out certain words, as this research focuses on the sentiment orientation of adjectives. One of the main conclusions from this research is that adjectives that are linked to each other with a conjunction like ‘and’ often have the same or at least a similar sentiment value. The opposite is true when adjectives are linked with ‘but’. Furthermore, Part-of-Speech can to some extent be used to detect negated information. Negations are crucial for proper sentiment analysis. People are more likely to use negations with negative sentiment than with positive sentiment, so positive words are negated to become negative, but negative words are usually not negated to get positive words [5].

In [5], the authors investigated Part-of-Speech patterns for sentiment analysis on Twitter data. For example, sequences such as “I just”, “I seriously”, “I never”, etc., are all patterns of the form ‘Personal Pronoun followed by Adverb’. In their research, this pattern proved to be associated with negative sentiment. The top 100 best patterns, ranked by their Information Gain score, is included as features, which significantly improves the performance compared to only using unigram features.

3 Lexico-Semantic Patterns

The various features we investigate in this research can be placed in six categories: synset, lexical, syntactical, sentiment, hybrid, and surface features. Synsets are a part of semantics, and are sets of synonyms that have a single meaning. Hence, synsets are more specific than the original words, since any ambiguity is eliminated. Both unigrams and bigrams are used here, with the caveat that for synset bigrams we ignore the order of the adjacent synsets. We believe this will make the features more robust, at only a small cost to accuracy. Hence, seeing synsets A and B is the same as seeing synsets B and A.

The lexical category consists of word patterns, where we use the lemma, or dictionary form, of each word in the patterns. We investigate unigrams through quadgrams, since n-grams with n larger than four are too sparse to be of practical use. The syntactical patterns are all sequences of Part-of-Speech (POS) labels. These labels match with any word in that particular word group (i.e., the ‘Noun’ label will match any noun). As such these patterns are more generic than lexical patterns, making them more robust, but less descriptive. We investigate POS patterns ranging from bigrams to quadgrams.

Furthermore, we look at negator-POS bigrams, which are in fact hybrids between syntactical and lexical features. The bigram consists of a negator, from a list of negator words like ‘not’ and ‘never’, followed by a Part-of-Speech label. It effectively splits the Part-of-Speech bigrams that start with an adverb into negating and non-negating bigrams, since words like ‘not’, ‘very’, and ‘highly’ all have the same Part-of-Speech label. We also look at hybrid patterns that combine a Part-of-Speech label and a synset in one bigram.

Since the task is sentiment classification, it makes sense to include sentiment related features as well. For that we use the SentiWordNet dictionary [1], where synsets are given a positive, negativity, and objectivity score that always sum up to one. We compute a sentiment score from those by subtracting the negativity score from the positivity score. This is denoted as a sentisynset. We also look at negator-sentisynset bigrams where a negator is followed by a sentisynset, since this will invert the influence it has on the sentiment classification.

The surface feature is actually not related to patterns, but instead it determines how much of the surrounding context in a sentence is taken into account when creating features for a given aspect. Whenever the exact location of an aspect within a sentence is provided in the annotated data, we use that to create a window of words around that aspect. The words within that window are the only source of information from which to create features for that specific aspect. This allows us to predict different sentiment classes for aspects that are in the same sentence. Unfortunately, for some aspects, the exact location within a sentence is not provided, in which case we cannot specify a specific window and are limited to use the whole sentence as a source for features. The window is defined as k words before the aspect and j words after the aspect, but bound to be within the same sentence.

4 Methodology

For the experiments, we use a linear multi-class Support Vector Machine (SVM). We perform a 10-fold cross-validation to ensure stable results, and from the 90 % training data, we designate 20 % as validation data. The latter is used to perform feature selection. The rest is used to train the SVM model itself. The final results, as reported in Table 5, are obtained by training on 80 % of the training data, using 20 % of the training data as a validation set, and evaluating on the official SemEval2015 test data for both data sets.

To determine which types of features perform the best, a forward feature selection is performed. In each round the effect of adding just an isolated feature type is measured. The feature type that gives the highest increase in performance is added to the selected set of features. Again, all remaining types of features are tested, until no increase in performance is measured. The baseline score is simply the majority class. For our data, the 'positive' sentiment class is the most prevalent, as can be seen in Table 1.

The two datasets that are used in our experiments are the English restaurant review data set and the English laptop review data set from SemEval 2015 [6].

The first part of the evaluation is dedicated to the feature selection, showing the effect of each type of feature on performance. First, starting with no features at all, the baseline always predicts positive (the majority class). Every type of

Table 1. Sentiment value distributions for the two used data sets.

	Positive	Neutral	Negative	Total
Restaurants	1198	53	403	1654
Laptops	1103	106	765	1974

Table 2. The effect of using an additional particular feature type versus the majority baseline for both data sets.

	Laptops	Restaurants
Baseline	0.497	0.637
+ *word unigram*	**0.754**	0.694
+ *word bigram*	0.738	**0.713**
+ *word trigram*	0.572	0.637
+ *word quadgram*	0.500	0.637
+ *POS bigram*	0.599	0.634
+ *POS trigram*	0.602	0.640
+ *POS quadgram*	0.525	0.637
+ *synset unigram*	0.696	0.669
+ *synset bigram*	0.597	0.672
+ *synset-POS bigram*	0.663	0.675
+ *negator-POS bigram*	0.555	0.637
+ *sentisynset unigram*	0.580	0.637
+ *negator-sentisynset bigram*	0.497	0.637

feature is added in isolation and the performance is measured. This is the first step in the forward feature selection and the results of this step are presented in Table 2. As expected, word unigrams and word bigrams are the two strongest types of features in this setup. Interestingly, the various feature types perform differently on the two data sets. Features that are useful for the laptop data are not beneficial for the restaurant data and the other way around. This shows how domain dependent sentiment analysis is.

Carrying out the forward feature selection procedure results in an optimal set of *word unigram*, *synset bigram*, *sentisynset unigram*, and *synset unigram* for the laptop domain and an optimal set of *word unigram*, *synset bigram*, *sentisynset unigram*, *POS bigram*, and *negator-POS bigram* for the restaurant domain.

Table 3. Results of the ablation experiments for both data sets. The '-' in the first column denotes set difference.

	Laptops accuracy	Restaurants accuracy
Using optimal feature set	76.80 %	73.18 %
- *word unigram*	−9.95 %	−0.99 %
- *synset bigram*	−2.49 %	−2.20 %
- *sentisynset unigram*	−1.94 %	−1.58 %
- *synset unigram*	−0.29 %	Not selected
- *POS-bigram*	Not selected	−2.21 %
- *negator bigram*	Not selected	−0.95 %

Reversing the above process is known as an ablation experiment. Here we start with the optimal set of features, and record the effect of removing one of the feature types. These results are shown in Table 3. Of interest is that while the word unigram features are very important for laptops, this is less true for restaurants, where synset bigrams and POS bigrams are the most important. In contrast, the sentisynset unigram feature is about as equally important for both domains.

Subsequently, the optimal window size is computed that limits the words from which features are extracted for a given aspect. This is only of interest for the restaurant data, since only there exact aspect locations are provided for many of the aspects. We find that the optimal window size is 8 words before and 8 words after the aspect (but always limited by sentence bounds). However, with $k = j = 7$ and $k = j = 9$, roughly the same performance is achieved, losing only 1.27 % in accuracy.

To go one level deeper, we looked at the weight of individual features as assigned in the trained SVM model. To make interpretation of these weights easier, we removed the 'neutral' class, resulting in a binary classifier (i.e. positive and negative only). Note that some words only appear with or even have just a single meaning. In that case, the (senti)synset feature has the same weight as the lexical feature of the same word (e.g., 'amazing' in the first column). Of interest are the domain specific words that appear with high weights, such as 'soggy' which is obviously negative for the restaurant domain, but is not used in the laptops domain, and 'Dell' which for this data set is an indicator of negative sentiment for laptops, but of course irrelevant for restaurants (Table 4).

The scores of the best performing feature sets for each data set are reported in Table 5. These use the optimal window size as discussed above. Overall, we obtain an F$_1$-score of 69 % for restaurant reviews and 73.1 % for laptops reviews. Looking at the precision and recall values for the different sentiment values, we can see that on the restaurant data, the SVM tends to classify too many aspects as positive, since both the precision for positive and the recall for negative is relatively low. This seems less the case for the laptops data, resulting in a higher overall score.

Table 4. The most influential features, according to the weight (positive or negative) assigned by the SVM. The feature types are denoted as folows: W is word unigram, SS is synset unigram, and SSS is sentisynset unigram. The SVM is run using the optimal set of feature types.

Restaurants				Laptops			
Positive		Negative		Positive		Negative	
Best (SSS)	0.348	Be (SSS)	−0.639	Be (SS)	0.893	Not (W)	−0.621
Be (SSS)	0.317	Not (SSS)	−0.562	Love (W)	0.696	Be (SS)	−0.593
Amazing (W)	0.31	Soggy (W)	−0.473	Amazing (W)	0.564	Worst (W)	−0.503
Amazing (SSS)	0.31	Worst (W)	−0.408	Great (W)	0.516	Worst (SS)	−0.503
Love (W)	0.304	Worst (SSS)	−0.408	Love (SS)	0.508	Dell (W)	−0.458

Table 5. Overview of classifications on the SemEval 2015 restaurants and laptops test data using the optimal features.

	Restaurants			Laptops		
	Precision	Recall	F$_1$-score	Precision	Recall	F$_1$-score
Positive	68.1 %	87.4 %	76.6 %	76.5 %	86.7 %	81.3 %
Neutral	33.3 %	4.4 %	7.8 %	22.2 %	10.1 %	13.9 %
Negative	72.7 %	53.2 %	61.4 %	72.6 %	66.0 %	69.1 %
All	69.0 %	69.0 %	69.0 %	73.1 %	73.1 %	73.1 %

5 Conclusion

In this work we employ and investigate lexico-semantic patterns for aspect-based sentiment analysis. We show that some of the investigated patterns improve the sentiment classification. For laptops the combination of word unigrams, synset unigram, synset bigrams, and sentisynset unigrams prove to be the best performing from amongst the feature types included in our experiments. It is interesting to see that semantical features such as synsets are preferred over other types of features such as the more syntactical Part-of-Speech (POS) bigrams. For restaurants, the best performing combination of feature types is word unigrams, synset bigrams, sentisynset unigram, POS-bigram, and negator-POS bigram. Again, the synset bigram is included, but additionally, the POS bigram and negator-POS bigram are included as well. Evidently, in the restaurant reviews, sentiment is expressed using more consistent syntactical patterns. This points to a difference in language use for reviews about laptops compared to reviews about restaurants. Exactly what these differences entail and why this phenomenon occurs is an interesting avenue for future research.

Another option for future work is to include even more feature types, as there are types of features that, as of yet, were not included in our experiments. Examples of these include additional lexicons and features based on grammatical relations. In conclusion, lexico-semantic patterns prove to be powerful predictors for sentiment analysis, as shown by the 69.0 % and 73.1 % F$_1$-score for restaurant and laptop reviews, respectively, but more research is needed to provide definitive answers.

Acknowledgments. The authors are partially supported by the Dutch national program COMMIT.

References

1. Baccianella, S., Esuli, A., Sebastiani, F.: SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010), vol. 10, pp. 2200–2204 (2010)

2. Brychcín, T., Konkol, M., Steinberger, J.: UWB: machine learning approach to aspect-based sentiment analysis. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 817–822. Association for Computational Linguistics and Dublin City University (2014)
3. Hatzivassiloglou, V., McKeown, K.R.: Predicting the semantic orientation of adjectives. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL 1997), pp. 174–181. Morgan Kaufman Publishers and Association for Computational Linguistics (1997)
4. Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S.: NRC-Canada-2014: detecting aspects and sentiment in customer reviews. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 437–442. Association for Computational Linguistics and Dublin City University (2014)
5. Koto, F., Adriani, M.: The use of POS sequence for analyzing sentence pattern in Twitter sentiment analysis. In: Proceedings of the 29th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA 2015), pp. 547–551. IEEE (2015)
6. Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., Androutsopoulos, I.: SemEval-2015 Task 12: aspect based sentiment analysis. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 486–495. Association for Computational Linguistics (2015)
7. Schouten, K., Frasincar, F.: The benefit of concept-based features for sentiment analysis. In: Gandon, F., Cabrio, E., Stankovic, M., Zimmermann, A. (eds.) ESWC 2015. CCIS, vol. 548, pp. 223–233. Springer, Heidelberg (2015)

Multilevel Browsing of Folksonomy-Based Digital Collections

Joaquín Gayoso-Cabada, Daniel Rodríguez-Cerezo,
and José-Luis Sierra(✉)

Fac. Informática, Universidad Complutense de Madrid,
C/Prof. José García Santesmases 9, 28040 Madrid, Spain
{jgayoso,drcerezo,jlsierra}@fdi.ucm.es

Abstract. This paper describes how to extend the usual one-level tag selection navigation paradigm in folksonomy-based digital collections to a *multilevel browsing* one, according to which it is possible to incrementally narrow down the set of selected objects in a collection by sequentially adding more and more filtering tags. For this purpose, we present a browsing strategy based on finite automata. As well, we provide some experimental results concerning the application of the approach in *Clavy*, a system for managing digital collections with reconfigurable structures in digital humanities and educational settings.

Keywords: Multilevel browsing · Folksonomy · Indexing · Navigation automata

1 Introduction

Folksonomies are cataloguing schemes defined and applied collaboratively by communities of users. In this way, users not only apply folksonomies to organize digital resources, but also actively contribute to their creation and maintenance [12]. In this context, accommodating any but the simplest interaction models can become a substantial technical challenge.

An example of a particularly difficult-to-achieve interaction model is general, unconstrained, *multi-level browsing* [5]. In this setting, users sequentially select tags, and, in each stage, the set of objects tagged by all the selected tags is filtered. Even for collections of moderate size, computing these sets of objects can in some cases be too costly to be achieved within acceptable response times. By establishing predefined orders in which tags can be selected and by using these orders to create and maintain navigation trees, response times can be dramatically enhanced, but this rigid and aprioristic organization is contrary to the dynamic and agile nature of folksonomies, where tag sets are continuously changing. In this paper we address this interaction style in its most unconstrained and general form.

The rest of the paper is organized as follows. Section 2 introduces the basis of folksonomy-like organizations of digital collections. Section 3 introduces the multilevel browsing paradigm for this kind of collections and describes how to enable such a browsing style efficiently. Section 4 presents some related work. Finally, Sect. 5 outlines the final conclusions and some lines of future work.

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 43–51, 2016.
DOI: 10.1007/978-3-319-48743-4_4

2 Folksonomy-Based Digital Collections

Collections organized with folksonomies typically comprise the following parts (see Fig. 1 for an example):

- On one hand, there are the *resources* in the collection. For instance, the small collection depicted in Fig. 1 includes six image archives as resources, corresponding to photographs of artistic objects from the Prehistoric and Protohistoric artistic periods in Spain (Fig. 1 actually shows thumbnails of these images).
- On the other hand, there is the *annotation* of the resources. This annotation consists of associating descriptive *tags* with resources. These tags are useful when cataloguing resources and, therefore, they enable future uses of the collection (navigation, search, etc.). For instance, in Fig. 1, resource number 1 has the tags *Cave-Painting*, *Cantabrian* and *Prehistoric* associated.
- Finally, there is a *tag cloud* that groups all the tags that can be used to annotate the resources. Thus, the tag cloud shown in Fig. 1 groups all the tags that annotate resources in the collection. As usual, the size of tags in this cloud represents the presence (number of tagged resources) of the tag in the collection.

Consequently, the internal organization of this kind of collection is very similar in appearance to classic keyword-based systems [15]. However, what distinguishes these collections from classic keyword-based systems is the social and inductive nature in the creation of the cataloguing schemata (i.e., the tag clouds). Indeed, folksonomy-based systems actively involve user communities that add, modify, delete and tag resources, using existing tags or creating new ones as needed. In this way, tag clouds are not explicitly defined nor explicitly maintained, but emerge from the collaborative behavior of communities of practice [12]. While this somewhat uncontrolled and anarchic approach to tagging digital resources can additionally bring up some relevant concerns and critiques from a cataloguing point of view (e.g., existence of synonymous, irrelevant or very generic tags, etc.) [14], the fact is that these systems are extensively used in many scenarios (and especially in computer-mediated social ones) [3]. Therefore, in this paper we will not focus on the critiques and potential shortcomings of the approach, but on efficient ways of enabling sophisticated interaction strategies (multi-level browsing, in particular).

Folksonomy-like systems support a simple one-level browsing strategy in a straightforward way. According to this strategy, it is possible to select one tag in the tag cloud and recover all the resources tagged with said tag. Figure 2(a) illustrates this approach with the small collection from Fig. 1.

One-level browsing can be accomplished efficiently in a straightforward way by using and maintaining an *inverted index* [19], i.e., a data structure that provides a reference to the set of resources tagged by each tag and therefore directly links to the results for each selection. This simple and efficient implementation explains why most folksonomy-based systems include this interaction style as a primary browsing strategy. However, this style prevents more sophisticated exploratory behaviors involving two or more tags simultaneously. In the rest of the paper we will examine how to deal with more than one browsing level.

Tag cloud

Tartesian
Megalithic
Plateau
Levant
Prehistoric
Punic
Protohistoric
Cave-Painting
Cantabrian
Phoenician
Penibaetic

Annotation

Resource 1	`Cave-Painting, Cantabrian, Prehistoric`	**Resource 4**	`Tartesian, Plateau, Protohistoric`
Resource 2	`Cave-Painting, Levant, Prehistoric`	**Resource 5**	`Phoenician, Penibaetic, Protohistoric`
Resource 3	`Megalithic, Cantabrian, Prehistoric`	**Resource 6**	`Punic, Levant, Protohistoric`

Resources

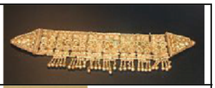
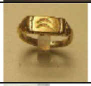
Resource 1		**Resource 4**	
Resource 2		**Resource 5**	
Resource 3		**Resource 6**	

Fig. 1. A small digital collection

3 Multilevel Browsing in Folksonomy-Based Systems

This section addresses the multi-level browsing style in folksonomy-based systems. Subsect. 3.1 introduces the basic interaction behavior. Subsect. 3.2 characterizes this behavior as a finite state machine. Finally, Subsect. 3.3 gives some experimental results.

3.1 The Browsing Model

Conceptually, the extension from one-level to multi-level browsing in folksonomy-like systems is simple. Basically, when a tag is selected, not only is the set of resources narrowed down but also the tag cloud: the resulting tag cloud will be the one *induced* by the set of filtered resources **R**. Such a tag will contain all the tags annotating some resource in **R** with the exception of those tags annotating *all* the resources in **R** (since, in this case, the selection would not refine the set of resources). This makes it possible

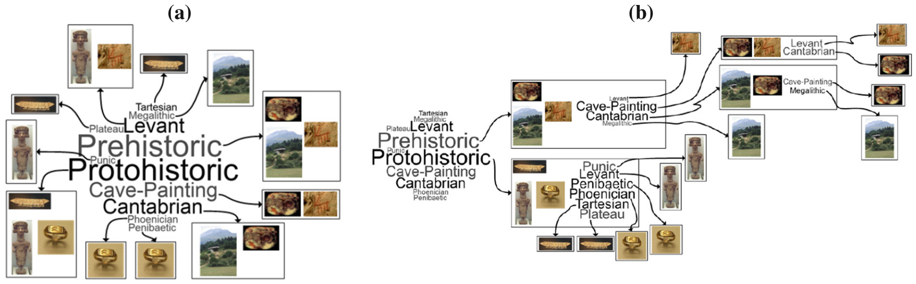


Fig. 2. Examples of (a) one-level browsing; (b) multi-level browsing

to carry out new selections successively on the narrowed tag clouds until a state containing an empty tag cloud is reached. The expected behavior is partially illustrated in Fig. 2(b), which shows the set of resources and the associated tag clouds after some browsing actions on the collection in Fig. 1.

As in the case of one-level browsing, multi-level browsing behavior can also be accomplished by using inverted indexes. However, now an evaluation of a conjunctive query in each interaction state is needed in order to determine the resources to be filtered. Although extensive research has been carried out on how to speed up these operations [2], in some cases the time inverted can negatively impact the user's interactive experience.

3.2 Navigation Automata

In order to accelerate multi-level browsing, it is necessary to have a suitable index structure. Ideally this structure should link to the set of resources selected by each meaningful set of tags t_1, …, t_n, in the same way, an inverted index directly provides the set of resources selected by a tag in the one-level approach. A way of providing such a structure is by using a finite state machine characterizing all the possible interactions and interaction states. This state machine will be called a *navigation automaton*. This automaton will consist of *states* labelled by sets of resources, and *transitions* labelled by tags (as an example, Fig. 3a shows the navigation automaton for the collection of Fig. 1). More precisely:

- There will be an initial state labelled by all the resources in the collection.
- Given a state **S** labelled by a set of resources **R**, for each tag **t** in the tag cloud induced by **R** there will be a state **S'** labelled by all the resources in **R** annotated by **t**, as well as a transition from **S** to **S'** labelled by **t**.

Since the navigation automaton contains all the possible ways of multi-level navigation, it can support multi-level browsing in a straightforward way. Unfortunately, in some cases the number of states in this automaton can grow very quickly (in the worst case, exponentially with respect to the number of resources). The most extreme case, in which the number of states is 2^n–1 (with n the number of resources), arises, for instance, by distinguishing each pair of resource annotations in a single tag. In order to

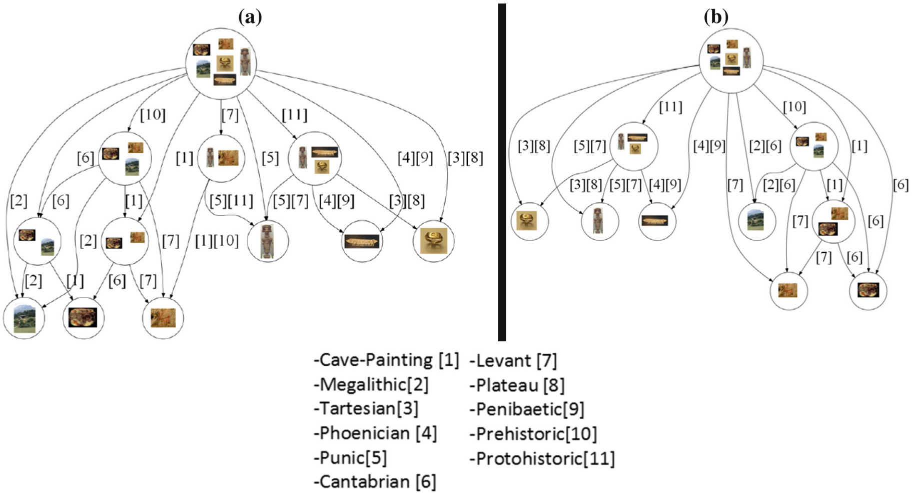


Fig. 3. (a) Navigation automaton for the collection in Fig. 1; (b) A non-deterministic version of the automaton in (a)

avoid this potential exponential factor in the explicit construction of navigation automata, it is possible to maintain non-deterministic versions of these automata, in such a way that only states representing disjoint partitions of their parent states are maintained. Figure 3b shows a feasible non-deterministic automaton equivalent to the one shown in Fig. 3a (it is worthwhile to point out that this solution may not be unique).

3.3 Experimental Evaluation

In order to evaluate our multilevel browsing approach, we have implemented it in *Clavy*, an experimental system for managing digital collections that lets users define organization schemata in a collaborative way.[1] In order to provide some structure to facilitate navigation, *Clavy* makes it possible to group tags in categories that are organized hierarchically. Nevertheless, this hierarchy is not pre-established, but can be edited by *Clavy* users at any time (see Fig. 4). Therefore, backstage, multi-level browsing support in *Clavy* must resort to the basic model described in Sect. 3, since the hierarchy is also subjected to continuous change and evolution. In addition to the automata-based browsing framework described in this paper, we have also implemented an inverted index-based solution in *Clavy*, using Lucene [13], a robust and highly optimized framework for implementing information retrieval applications.

In this context, we set up an experiment consisting of adding the resources in *Chasqui* [17],[2] a digital collection of 6283 digital resources on Pre-Columbian

[1] http://clavy.fdi.ucm.es/Clavy/.

[2] http://oda-fec.org/ucm-chasqui.

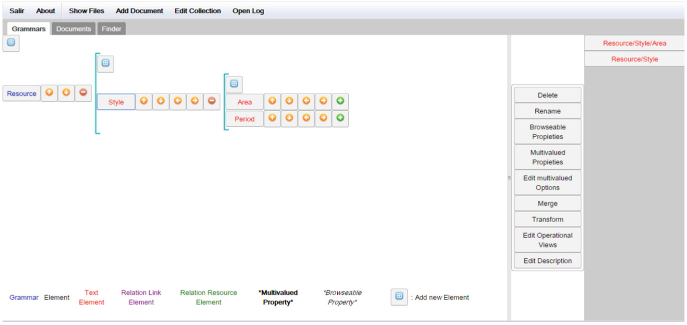


Fig. 4. Editing a hierarchy of tag categories with *Clavy*.

American archeology, to *Clavy* and simulating runs concerning hierarchy reconfiguration and browsing operations.

Each run was customized as follows. We interleaved resource insertion with hierarchy reconfiguration/browsing rounds. Each insertion round consisted of 100 resource insertions (with the exception of the last one, in which all the remaining resources where inserted). In turn, each browsing/reconfiguration round consisted of executing $0.1n$ browsing operations randomly interleaved with $0.01n$ reconfigurations (n being the number of resources inserted so far). Each browsing operation in turn consisted of selecting a feasible tag and computing the next set of active objects, or of establishing the initial state as the active one in case of unavailability of feasible tags; once the next interaction state was determined, all the filtered resources were visited. In both the cases of inverted indexes and automata, in-memory indexes were used in order to avoid the side effects of persistence that might disturb the experiment.

Figure 5 shows the results obtained from the two runs. The experiment was run on a PC with Windows 10, with a 3.4 GHz Intel microprocessor, and with 8 Gb of DDR3 RAM. The horizontal axis corresponds to the number of operations carried out so far. The vertical axis corresponds to cumulative time (in seconds). As is made apparent, the automata-based approach clearly outperforms inverted indexes (regardless of the fact that we are using a highly optimized framework, like Lucene, for inverted indexing vs. our own in-house experimental implementation for navigation automata).

4 Related Work

There are several systems that, like our proposal, implement several sorts of multi-level browsing onto folksonomy-based systems. Systems like the one described in [7, 9] are supported by inverted index approaches. Other systems, like that described in [11], are supported by extensible data adapters that interface between synchronized tag clouds and underlying database management systems. Instead of relying on inverted indexes and/or conventional database layers, our approach starts by characterizing the intrinsic behavior of multi-level browsing onto a folksonomy-like system in terms of navigation

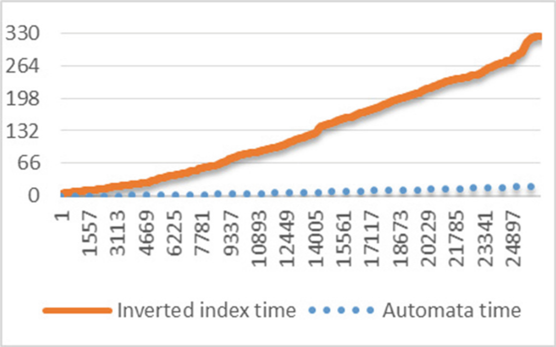


Fig. 5. Cumulative time of inverted indexes vs. automata

automata, and then tries to approximate this model with a non-deterministic version that provides reasonable time and space tradeoffs. In [4] we propose a representation of these non-deterministic automata inspired by *dendrograms* such as those used in hierarchical clustering settings [8].

Our navigation automata model is actually similar to lattice-based proposals to browse information spaces, as described in the seminal work of [5]. This organization is actually the main subject of the fertile theory of *formal concept analysis* [16]. Similarly, there are several proposals on using lattices as the underlying indexing structures for enabling multi-level browsing [6, 18]. However, all these approaches are limited by the intrinsic complexity of formal concept analysis [10]. This is why we have proposed a simpler but still practical approximation based on non-deterministic versions of navigation automata.

5 Conclusions and Future Work

Folksonomy-based digital collections are living entities in which not only digital resources, but also organization schemata, are subject to continuous change and evolution. This changing and evolving nature makes the accomplishment of sophisticated interaction paradigms particularly challenging. In this paper we have addressed the efficient inclusion of multilevel browsing strategies in these settings, in which sets of selected resources can be successively refined through the selection of sequences of tags. For this purpose we have modeled this behavior as a finite state machine, the *navigation automaton*, taking into account all the possible ways of navigating the collection by using tags. Unfortunately, we have also showed how, in some cases, the number of states in this automaton can increase exponentially with respect to the collection's size. In order to address this potential exponential factor we have proposed using non-deterministic versions of these automata. Some experiments with a real collection gave us evidence on how the automata-based technique can outperform more conventional and widely used ones, like those based on inverted indexes.

We are currently working on further optimizing our navigation automata representation. We are also looking for efficient ways to make all this information persistent, either by using standard relational databases or alternative NoSQL approaches. Finally, we also hope to include support for arbitrary Boolean queries and for alternative ways of exploring the resources selected.

Acknowledgements. This work has been supported by the BBVA Foundation (grant HUM14_251) and Spanish Ministry of Economy and Competitiveness (grant TIN2014-52010-R)

References

1. Chodorow, K.: MongoDB: The Definitive Guide. O'Reilly, Sebastopol (2013)
2. Culpepper, J-S., Moffat, A.: Efficient set intersection for inverted indexing. ACM Trans. Inf. Syst. **29**(1) (2010)
3. du Preez, M.: Taxonomies, folksonomies, ontologies: what are they and how do they support information retrieval? Indexer **33**(1), 29–37 (2015)
4. Gayoso-Cabada, J., Rodríguez-Cerezo, D., Sierra, J-L.: Browsing digital collections with reconfigurable faceted thesauri. In: 25th International Conference on Information Systems Development (ISD), Katowize, Poland (2016)
5. Godin, R., Saunders, G.: Lattice model of browsable data space. Inf. Sci. **40**(2), 89–116 (1986)
6. Greene, G-J.: A generic framework for concept-based exploration of semi-structured software engineering data. In: Proceedings of the 30th IEEE/ACM International Conference on Automated Software Engineering, pp. 894–897 (2015)
7. Hernandez, M-E., Falconer, S-M., Storey, M-A., Carini, S., Sim, I.: Synchronized tag clouds for exploring semi-structured clinical trial data. In: Proceedings of the 2008 Conference of the Center for Advanced Studies on Collaborative Research: Meeting of Minds (CASCON), article 4 (2008)
8. Jain, A.-K., Murty, M.-N., Flynn, P.-J.: Data clustering: a review. ACM Comput. Surv. **31** (3), 264–323 (1999)
9. Koutrika, G., Zadeh, Z-M., Garcia-Molina, H.: CourseCloud: summarizing and refining keyword searches over structured data. In: Proceedings of the 12th International Conference on Extending Database Technology (EDBT), pp. 1132–1135 (2009)
10. Kuznetsov, S.: On computing the size of a lattice and related decision problems. Order **18**(4), 313–321 (2001)
11. Leone, S., Geel, M., Müller, C., Norrie, M.C.: Exploiting tag clouds for database browsing and querying. In: Proper, E., Soffer, P. (eds.) CAiSE Forum 2010. LNBIP, vol. 72, pp. 15–28. Springer, Heidelberg (2011)
12. Mathes, A.: Folksonomies – cooperative classification and communication through shared metadata. Comput. Mediat. Commun. – LIS590CMC **47**(10), 1–13 (2004)
13. McCandless, M., Hatcher, E., Gospodnetic, O.: Lucene in Action, 2nd edn. Manning Publications, Greenwich (2010)
14. Peterson, E.: Beneath the metadata: some philosophical problems with folksonomy. D-Lib Mag. **12**(11) (2006)
15. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Maidenherd (1986)

16. Sarmah, A.-K., Hazarika, S.-M., Sinha, S.-K.: Formal concept analysis: current trends and directions. Artif. Intell. Rev. **44**(1), 47–86 (2015)
17. Sierra, J.-L., Fernández-Valmayor, A., Guinea, M., Hernanz, H.: From research resources to learning objects: process model and virtualization experiences. Educ. Technol. Soc. **9**(3), 56–68 (2006)
18. Way, T., Eklund, P.: Social tagging for digital libraries using formal concept analysis. In: Proceedings of the 17th International Conference on Concept Lattices and their Applications (CLA) (2010)
19. Zobel, J., Moffat, A.: Inverted files for text search engines. ACM Comput. Surv. **33**(2) (2006). Article 6

Ranking in Social Networks

Faderank: An Incremental Algorithm for Ranking Twitter Users

Massimo Bartoletti[1(✉)], Stefano Lande[1], and Alessandro Massa[1,2]

[1] Università degli Studi di Cagliari, Cagliari, Italy
bart@unica.it
[2] Xorovo.com, Cagliari, Italy

Abstract. User reputation is a crucial indicator in social networks, where it is exploited to promote authoritative content and to marginalize spammers. To be accurate, reputation must be updated periodically, taking into account the whole historical data of user activity. In big social networks like Twitter and Facebook, these updates would require to process a huge amount of historical data, and therefore pose serious performance issues. We address these issues in the context of Twitter, by studying a technique which can update user reputation in constant time. This is obtained by using an arbitrary ranking algorithm to compute user reputation in the most recent time window, and by combining it with a summary of historical data. Experimental evaluation on large datasets show that our technique improves the performance of existing ranking algorithms, at the cost of a negligible degradation of their precision.

Keywords: Reputation · Social networks · Performance

1 Introduction

The global growth of electronic communication facilities like peer-to-peer and social networks brought out two major problems: how to filter out low quality information, and how to enforce safe interactions. In many contexts, these issues are not completely disjoint: e.g., interacting safely in peer-to-peer networks may correspond to only download trusted contents or files. A possible way to address these issues is to associate each object (peer, user or resource) with an index, usually called *reputation*, which reflects the opinion the network has towards such object. The underlying assumption is that, by analysing the past interaction history of an object, one can predict the quality of its future interactions [10,22].

In the specific context of social networks, like e.g. Twitter and Facebook, reputation has several applications: for instance, it has been exploited to rank users [1,21], to marginalize spammers [26] and dishonest services [5], to distributed moderation among users [13,14], to maximize information spread in viral marketing strategies [11], and to refine search results [18].

Designing effective reputation systems for social networks is not an easy task, for two main reasons. First, they have to deal with the impressive amount of data generated by social networks: for instance, Twitter counts ∼10 M daily active

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 55–69, 2016.
DOI: 10.1007/978-3-319-48743-4_5

users and ∼500 M tweets per day [2], while Facebook counts ∼900 M daily active users, ∼44 M comments and ∼4.5 B likes per day [3]. Second, reputation system must protect themselves from misbehaving users who try to undermine the ranking mechanism in order to obtain unwarranted service or to prevent honest participants from obtaining legitimate service [8]. Although some defence techniques against these attacks have been proposed over the years [6,17,23,29,30], currently there is no reputation system which is (either provably or empirically) resilient to all kinds of attacks.

To make the situation even more complex, the problem of efficiency and that of security are strictly related. On the one hand, if a system tries to improve efficiency by reducing the frequency of reputation updates, an adversary could easily build a positive reputation in a first period of time, and then exploit it to carry on attacks in the time window where reputation is not updated. On the other hand, frequently recomputing reputation gives rise to a performance problem: ideally, for each update we have to process all the historical data, besides the new data. A possible approach to mitigate this issue is to truncate data older than a certain time: for instance, Klout—a popular reputation aggregator—only considers the last 90 days of user interaction [21]. However, this mechanism can be subject to *whitewashing attacks* where an adversary abuses the system for a while, and then simply waits some time before rebuilding a fresh reputation.

Contributions. In this paper we propose and evaluate a technique to reduce the overhead of keeping updated the reputation of Twitter users. Instead of naïvely truncating historical data, our technique aggregates it in constant time and space, by adapting the *fading memories* technique of [23]. The actual reputation of a user is computed by taking into account its recent (raw) behaviour, its behaviour in the aggregated history, and the gradient of behaviour change. In this way, we reach two goals. First, since the amount of data to be processed at each update is (on average) constant, the average execution time of an update is constant as well. Second, since the past interaction history is taken into account (although in aggregated form), we mitigate *whitewashing attacks* like the ones outlined above. A further feature of our technique is that it is parametric with respect to the reputation algorithm used to process raw data. Overall, one can choose the algorithm which offers the required defences on raw reputation, and calibrate the weights of the raw/aggregated/gradient components to obtain similar properties on the optimized algorithm.

We validate our technique, called *Faderank*, using two raw reputation algorithms: TURank [28], and a variant of PageRank [20] suited to rank Twitter users. In our experiments we use three real datasets, obtained by crawling Twitter for several weeks. Our datasets contain tweets, retweets, and follow relations of ∼10 K users, spanning over a period of eleven months. Assuming a monthly reputation update, we compare the completion time of FadeRank, TURank and PageRank, showing that Faderank is a constant-time algorithm, while the computation time of the raw algorithms grows linearly on the size of the input. To evaluate the precision of Faderank we use the *Kendall τ rank distance* [12]. More precisely, for each dataset, for each iteration, and for each raw algorithm (TURank and PageRank), we measure two distances: the distance between the ranking obtained by Faderank and the raw algorithm, and the distance between the latter and its *forgetful* version, where the history is truncated every month. Our experiments show that, in all datasets, there is a little degradation of the precision of Faderank w.r.t. the raw algorithms, but Faderank is still more precise than their forgetful versions. Further, we show that, compared with the forgetful algorithms, Faderank is more resilient to whitewashing attacks.

The sources of our FadeRank tool, as well as the experimental data used for its validation, are available online at tcs.unica.it/software/faderank.

2 Related Work

In this section we briefly survey the literature on reputation systems, with special emphasis on those used for ranking users of social networks.

Pagerank. Many reputation systems are based on PageRank [20], an algorithm originally introduced by Google to rank web pages. PageRank models the web as a directed graph (V, E), where V is the set of web pages, and E is the set of hyperlinks (i.e., references) from a page to another. The reputation of a web page is proportional to the reputation of the web pages that reference it. Being based on a single object-object relation, the PageRank model does not precisely capture the rich topology of social networks. In Twitter, for instance, users can follow other users, and send "tweets" which can be "retwitted" by other users. Designing a reputation system which flattens this structure to a single user-user relation may affect the precision of the results; hence, subsequent works have refined the PageRank model to take into account more complex structures.

ObjectRank. ObjectRank [4] generalises the PageRank model by considering different kinds of edges and nodes. Each node gives a part of its reputation to the nodes linked to it. The exact amount of this reputation is determined by (i) the weight on the edge which links the two nodes, and (ii) the reputation of the source node. To account for the fact that different kind of relations may affect the reputation in different ways, ObjectRank allows to associate a different weight to each kind of edge.

To apply ObjectRank to a new domain, one has to instantiate an *authority transfer schema graph* (V_S, E_S, w_S), where V_S is the set of *node kinds*, E_S is the set of *edge kinds*, and $w_S : E_S \to \mathbb{R}$ associates weights to edge kinds. From this schema graph and the dataset, ObjectRank constructs an *authority transfer graph* (V, E, w, k), which is used to compute the reputation. The component V is the set of nodes (the actual objects in the dataset), while E is the set of edges (associated to an edge kind by $k : E \to E_S$). The component $w : E \to \mathbb{R}$ associates a weight to each edge as follows:

$$w(e) = \frac{w_S(k(e))}{OutDeg_{k(e)}(u)} \tag{1}$$

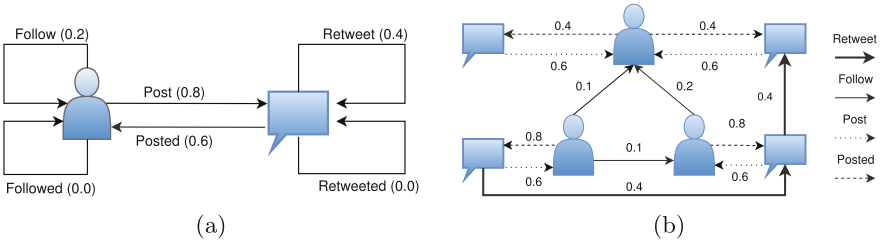


Fig. 1. In 1a, the user-tweet schema graph; in 1b, a user-tweet graph.

where $OutDeg_{e_S}(u) = \#\{e \in E \mid fst(e) = u \text{ and } k(e) = e_S\}$ is the number of edges of type e_S originating from the node u.

Similarly to PageRank, the reputation is a vector $\boldsymbol{r} \in \mathbb{R}^{|V|}$, defined as the fixed point of the following equation:

$$\boldsymbol{r} = d\, \boldsymbol{A}\, \boldsymbol{r} + \frac{(1-d)}{|V|}[1, \ldots, 1]^T \tag{2}$$

where d is a real constant (called *damping factor*), and $\boldsymbol{A} = (a_{uv})$ is the *transition matrix* where each element a_{uv} is given by

$$a_{uv} = \begin{cases} w((u,v)) & \text{if } (u,v) \in E \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

ObjectRank computes $\boldsymbol{r}$ through an iterative algorithm, which converges whenever the transition matrix $\boldsymbol{A}$ is irreducible and aperiodic [19]. The first condition is guaranteed by a suitable choice of the damping factor d, while the second one happens to be true for real-world datasets.

TURank. The reputation system proposed in [28], called TURank, instantiates the authority transfer schema graph of ObjectRank into a *user-tweet schema graph* $UTG_S = (V_S, E_S)$, displayed in Fig. 1a. The set of nodes V_S comprises just two elements (*user* and *tweet*). The set of edges E_S renders the fact that Twitter users can: (i) dispatch *tweets*, (ii) *follow* users (i.e., when a user A follows B, she will receive all tweets posted by B), and (iii) *retweet* the messages they receive (i.e., if B retweets a message of A, this message will be received by all the followers of B). The weights associated to edges reflect the following assumptions: (i) authoritative users tend to follow other authoritative users; (ii) tweets retweeted by many authoritative users are likely to be interesting; (iii) users who post many interesting tweets are likely to be authoritative.

The authority transfer graph instantiated from the schema and the dataset is called *user-tweet graph*, a minimalistic example of which is displayed in Fig. 1b. TURank analyses this graph, using Eq. (2) to compute the reputation of Twitter users. Note that, as anticipated in Sect. 1, to update the reputation one must considering both historical data and new data to reconstruct the user-tweet graph and analyse it. We discuss in Sect. 3 how our proposal reduces the computational overhead of this operation.

Other Adapations of Pagerank. Several other papers propose reputation systems for social networks by taking inspiration from PageRank and ObjectRank. Weng *et al.* [27] propose an algorithm also takes into account the topic similarity between the users (i.e., two users are similar to the extent they tweet on similar topics). More precisely, the algorithm in [27] can associate to each user many reputation values, i.e. one for each topic. To this aim, the PageRank model is modified in [27] so to have a different transition probability between each node (i.e. the values of the transition matrix), depending on the topic similarity between them. Haveliwala [7] propose a similar algorithm to improve PageRank using topics, but they use a different "teleportation" vector (the $[1, \ldots, 1]^T$ vector in Eq. (2)) for each topic, instead of changing the transition matrix.

Time-Sensitive Algorithms. Mariani *et al.* [16] investigate the problem of how temporal aspects affect reputation systems based on PageRank, reaching the conclusion that not considering these aspects undermines their accuracy. Hu *et al.* [9] address this issue in the field of social networks, by adapting PageRank to take into account three time factors: the age of an edge, the frequency with which edges are created, and the topic similarity of the nodes linked by new edges in a certain amount of time, under the assumption that trustworthy users focus their activity in a certain topic for a period of time.

Algorithms Based on Other Techniques. Many works propose reputation systems for social networks which do *not* employ the link-structure analysis with the PageRank model, and exploit instead machine learning techniques. Uysal and Croft [24] compute a reputation for the incoming tweets of a user. The reputation of a tweet is proportional to the probability that the recipient will retweet it, and it is computed using a decision tree. To do that, they propose several features, like e.g. the number of followers of the author, the number of retweets, and the contents of the tweet itself. Wang [26] proposes a similar approach, using a Bayesian classifier to detect spam tweets. Differently from [24], the algorithm in [26] uses features obtained from a graph similar to the user-tweet graph of TURank. Vosecky *et al.* [25] propose a filter model that uses a SVM classifier to discard low quality tweets, and a rank model that uses Rank SVM to order tweet by reputation. They use two sets of features: content-based (like e.g. punctuation, spelling, grammatical indicators), and link-based, that utilizes the implicit relations between tweets, hyper-links, and users. Ma *et al.* [15] associates a reputation to tweets, rather than users. The reputation of a tweet is a measure of its popularity, and it is computed using a *sigmoid function*, taking into account the number of retweets, the number of possible views (i.e., number of user that can see the tweet because they follow the author or other users that retweeted it), and a model of the temporal dynamics of a tweet.

3 FadeRank

In this section we illustrate our technique, which is obtained by suitably combining three basic ingredients:

- an *arbitrary ranking algorithm*, used to compute the *raw reputation* from the data collected in a time interval;
- the *dependable trust model*, used to compute the *aggregated reputation* from the raw reputation and the historical data;
- the *fading memories* technique, used to aggregate the historical data in constant time and space.

Overall, we call FadeRank the combination of these three ingredients. Before introducing in Sect. 3.3 our algorithm, we present in Sects. 3.1 and 3.2 the dependable trust model and the fading memories technique.

3.1 Dependable Trust Model

We exploit the dependable trust model of [23] to compute the *aggregated reputation* of users, taking as input the raw reputation obtained by an arbitrary ranking algorithm. The aggregated reputation of a given user at time interval $i \geq 0$, denoted by $AR[i]$, is the weighted sum of three components:

$$AR[i] \quad = \quad \alpha \cdot R[i] \ + \ \beta \cdot H[i] \ + \ \gamma(i) \cdot D[i] \tag{4}$$

where:

- $R[i]$ is the raw reputation of the user at time interval i;
- $H[i]$, defined by Eq. (5) below, aggregates in a single value the *history* of the user reputation over the time intervals $0, \dots, i-1$;
- $D[i]$, defined by Eq. (6) below, represents the change of reputation of the user in the last time interval.

More precisely, the value $H[i]$ is computed as follows:

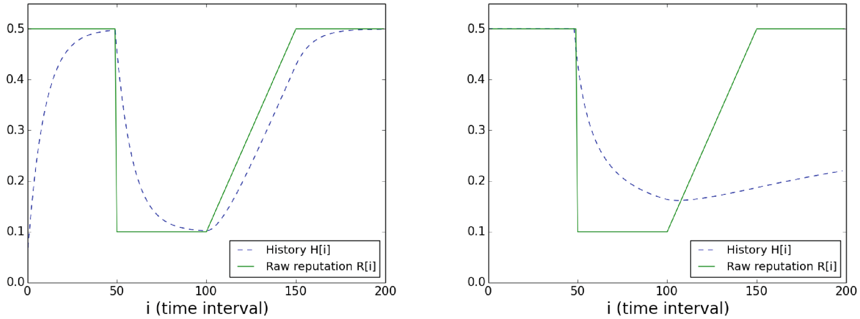
$$H[i] \quad = \quad \sum_{k=1}^{n} R[i-k] \cdot \frac{w_k}{\sum_{j=1}^{n} w_j} \qquad \text{where } n = \begin{cases} maxH & \text{if } i \geqslant maxH \\ i & \text{otherwise} \end{cases} \tag{5}$$

and where $maxH$ is the number of past raw reputation values stored by the system, and w_k is a weight. Some possible choices for w_k (taken from [23]) and their effects are displayed in Fig. 2.

The value $D[i]$ is computed as the difference between the aggregated history and the current raw reputation as follows:

$$D[i] = R[i] - H[i] \tag{6}$$

The weights α, β, and γ in Eq. (4) can be tuned to change the response of the reputation system to attacks. A larger value of α gives more weight to the



(a)

(b)

Fig. 2. The figures show a simulation of a whitewashing attack. We compare the trend of the history values $H[i]$ (solid lines) to the raw reputation values $R[i]$ (dashed lines), i.e. the behaviour of the attacker, for two different choices of w_k. Figure 2a shows an *optimistic* choice, i.e. $w_k = \rho^{k-1}$ (with $\rho < 1$), for which Eq. (5) becomes an exponentially weighted sum. Instead, Fig. 2b shows a *pessimistic* choice, i.e. $w_k = \frac{1}{R[i-k]}$, which yields a harmonic mean.

recent behaviour, while a larger value of β gives more weight to the past history (to address e.g., whitewashing attacks [10]).

The weight $\gamma(i)$ at time interval i is given by:

$$\gamma(i) = \begin{cases} \gamma_1 & \text{if } D[i] \geqslant 0 \\ \gamma_2 & \text{otherwise} \end{cases} \tag{7}$$

where γ_1 and γ_2 are two constants. A possible choice of these constants, as suggested by [23], could be $\gamma_1 < \beta < \gamma_2$. In the first case ($D[i] \geq 0$) we reward by a factor γ_1 an amelioration of the user behaviour, while in the second one ($D[i] < 0$) we penalise by a factor γ_2 its deterioration.

3.2 Fading Memories

When computing the history value $H[i]$ in Eq. (5), we assume that the system stores the raw reputation values for a user for the past *maxH* intervals. If, to account for the distant past, we were storing a large number *maxH* of values, we would cause a large memory footprint, as well as an increase of the time needed to compute $H[i]$ upon each update. On the other hand, a small value of *maxH* would make the malicious behaviour of a user forgotten after *maxH* intervals, so paving the way to whitewashing attacks.

To cope with this issue, we exploit the *fading memories* technique of [23], which allows to compute a bounded digest of the *whole* past history, and so to compute the value $H[i]$ in constant time and space. To do that, we store only the most recent raw reputation values exactly, while we aggregate (*fade*) the older values, with an accuracy that decreases proportionally to their age.

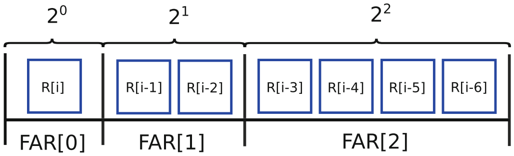


Fig. 3. Fading memories with $b = 2$ and $m = 3$, which aggregate $b^m - 1 = 7$ past raw reputation values into m values. The faded values $FAR[i]$ (for $i \in 0 \dots 2$) are obtained by aggregating $b^0 = 1$, $b^1 = 2$, and $b^2 = 4$ past raw reputation values.

More precisely, fixed two strictly positive integer constants b and m, the fading memories aggregate into m values the past reputation values: the i-th value is the digest of b^i reputation values. The 0-th fading memory is a digest of the b^0 most recent reputation value (i.e., just $R[i]$), the 1-th is a digest of b^1 values (i.e., $R[i-1], \dots, R[i-b]$), and so on. Since $\sum_{i=0}^{m-1} b^i = b^m - 1$, the m fading memories actually represent a digest of the last $b^m - 1$ reputation values. Figure 3 shows an example of fading memories with $b = 2$ and $m = 3$. The memories for the recent past aggregates a smaller number of raw reputations than the one for the old past, thus making the former more precise.

To update the fading memories at the stroke of a new time interval we use Eq. (8) below, where we denote with $FAR^t[i]$ (for $0 \leq i \leq m - 1$) the i-th faded past reputation value at interval t of a given user:

$$FAR^{t+1}[i] \quad = \quad \frac{FAR^t[i] \cdot (b^i - 1) \; + \; FAR^t[i-1]}{b^i} \tag{8}$$

3.3 The FadeRank Algorithm

Our FadeRank algorithm is illustrated in Algorithm 1. The main routine is FadeRank (lines 1–7), which takes as input *newData*, the user-tweet graph corresponding to the interactions in the most recent time interval. This graph is then passed to a ranking algorithm (line 2), which computes the raw reputations on *newData* (i.e., the ranking algorithm considers *newData* as the whole history of interactions). The call to ComputeReputation (line 4) updates the reputation of a user, exploiting the dependable trust model described in Sect. 3.1.

The function ComputeReputation takes as input the current fading memories vector *far*, and the *score* for the interval computed before. At line 9 we aggregate the *far* values into a single value to compute *history*, according to Eq. (5). The *score* and *history* values are then used to compute the change of reputation *diff* (line 10) according to Eq. (6). At line 11 we compute the weight γ as in Eq. (7), and then at line 12 we compute the new reputation of the user, exploiting Eq. (4). The last step is the update of *far* (line 5), performed by the function UpdateFAR, that employs Eq. (8) (line 16).

Algorithm 1. FadeRank

```
procedure FADERANK($newData$)
    $rawScores \leftarrow$ REPUTATIONALGORITHM($newData$)
    for $i$=1 to $|users|$ do
        $users_i.score \leftarrow$ COMPUTEREPUTATION($users_i.far$, $rawScores_i$)
        $users_i.far \leftarrow$ UPDATEFAR($users_i.far$, $rawScores_i$)
    end for
end procedure

function COMPUTEREPUTATION($far$, $score$)
    $history \leftarrow \sum_{i=1}^{|far|} far_i \cdot \frac{w_i}{\sum_{k=1}^{|far|} w_k}$
    $diff \leftarrow score - history$
    $\gamma \leftarrow$ if $diff \geqslant 0$ then $\gamma_1$ else $\gamma_2$
    return $\alpha \cdot score + \beta \cdot history + \gamma \cdot diff$
end function

function UPDATEFAR($far$, $score$)
    for $i$=1 to m do
        $newFAR_i \leftarrow \frac{far_i \cdot (b^i - 1) + far_{i-1}}{b^i}$
    end for
    return $newFAR$
end function
```

4 Validation

In this section we validate FadeRank in terms of its performance and precision with respect to two raw ranking algorithms, i.e. TURank [28] and a variant of PageRank [20] tailored to Twitter[1]. Additionally, we compare the precision of both instances of FadeRank with the *forgetful* versions of the raw ranking algorithms, which truncate the history every month.

Hereafter, we shall denote with:

- FadeRank$_{\mathbf{T}}$: the instance of FadeRank which uses TURank as source of raw reputation;
- FadeRank$_{\mathbf{P}}$: the instance of FadeRank which uses our variant of PageRank;
- Forget$_{\mathbf{T}}$: the forgetful version of TURank;
- Forget$_{\mathbf{P}}$: the forgetful version of PageRank.

Datasets. To the purpose of validation we have constructed three datasets, obtained by a custom crawler which downloads data (i.e., the tweet, retweet

[1] Note that we cannot use PageRank *as is* because of limitations of Twitter APIs, which do not allow to obtain temporal information about the "follow" relation. To circumvent this limitation, our variant of PageRank operates on the "tweet" and "retweet" relations, by assigning to each user the sum of the score of its tweets.

and follow relations) exploiting the Twitter APIs. Table 1 shows the details of the datasets; all of them cover a time-span of 11 months. The datasets D1 and D3 contain data of Italian users (the former has a relevant portion of influential users; the latter contains mostly normal users), while the dataset D2 contains data of American users.

In the rest of this section we present the validation results only for dataset D1; the analysis of the other datasets leads to very similar results, so to save space we make it available online at tcs.unica.it/software/faderank.

Table 1. Datasets details.

Dataset	#Users	#Tweet	#Follow	#Retweet
D1	∼11 K	∼14 M	∼15 M	∼5 M
D2	∼12 K	∼12 M	∼15 M	∼4 M
D3	∼12 K	∼11 M	∼11 M	∼3 M

Partitioning the Datasets in Time Intervals. We are interested in evaluating and relating the performance and the precision of *incremental* algorithms (i.e. FadeRank$_{\mathbf{T}}$, FadeRank$_{\mathbf{P}}$, Forget$_{\mathbf{T}}$, and Forget$_{\mathbf{P}}$) with respect to *non-incremental* ones (i.e., TURank and PageRank). To this purpose, we partition each dataset in 11 time intervals, each one comprising data spanning over 30 days. Then, we execute the algorithms to update user reputation, with the following criteria:

- incremental algorithms: for each time interval, process only the data contained in such interval;
- non-incremental algorithms: for each time interval i, process the data from the first to the i-th time interval.

Choice of the Parameters. For the purpose of the validation, we have to fix actual values for several parameters:

- the weights α, β, γ_1, γ_2 of the function AR in Eqs. (4) and (7);
- the weight function w_k in Eq. (5);
- the values for the parameters b and m of the fading memories (Eq. (8)).

The choice of the parameters for the dependable trust model aims at equalizing the scale and behaviour of FadeRank and of the raw reputation algorithm (TURank and PageRank). To this purpose, we choose $w_k = \rho^{k-1}$ as in Fig. 2a (with $\rho = 0.9$), and we compute the weights α, β, γ_1, γ_2 as follows:

1. we start by executing FadeRank$_{\mathbf{T}}$ (resp. FadeRank$_{\mathbf{P}}$) on dataset D1, simulating an update of the user reputation in 30-days intervals;
2. for each update, we save the raw reputation (denoting with R_t^n the raw reputation of user n at interval t) and the *history values* (denoting with H_t^n the history value of user n at interval t);

3. we execute TURank (resp. PageRank) with the data from interval 0 to t, so to compute the reputation of each user (denoting with T_t^n the reputation of user n at the interval t);
4. we solve the over-determined linear system obtained with the equations in the form $\alpha \cdot R_t^n + \beta \cdot H_t^n = T_t^n$, using the minimum least squares method;
5. finally, we use the solution of the linear system as initial values of α and β, which we fine-tune to mimic the behaviour of TURank (resp. PageRank); using the same criteria we choose the parameters γ_1 and γ_2[2].

For the fading memories parameters we choose $b = 2$ and $m = 3$, so to have a small number (i.e., $2^3 - 1 = 7$) of values to store. Note that, by increasing the number of fading memories (i.e., choosing bigger values for b and m), the FadeRank algorithm would take account for the past more precisely. Table 2 summarizes the choice of parameters used in the validation.

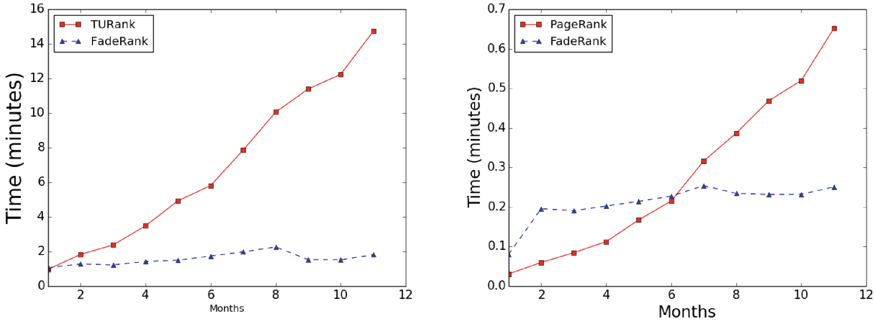
Table 2. Choice of the parameters of FadeRank$_\mathbf{T}$ and FadeRank$_\mathbf{P}$.

Algorithm	α	β	γ_1	γ_2	w_k	ρ	b	m
FadeRank$_\mathbf{T}$	0.3	0.9	0.1	0.1	ρ^{k-1}	0.9	2	3
FadeRank$_\mathbf{P}$	0.3	1.2	0.1	0.1	ρ^{k-1}	0.9	2	3

Performance Analysis. Figure 4 shows the execution time of FadeRank$_\mathbf{T}$ *vs.* TURank (Fig. 4a), and of FadeRank$_\mathbf{P}$ *vs.* PageRank (Fig. 4b). As expected, the experimental results show that the execution time of the non-incremental algorithms grows linearly with time (because the amount of data to be analysed grows at each update), while the execution time of FadeRank remains more or less constant. Note that the execution time of FadeRank cannot be exactly constant, because the size of the partition of the dataset is not constant (e.g., the monthly number of tweets may vary). The execution time of the forgetful versions of the algorithms (not shown in the figure) are very close to that of FadeRank.

Precision Analysis. To evaluate the precision of FadeRank, we consider *rankings*, i.e. list of Twitter users sorted by their reputation (computed each month). More precisely, we compare the ranking of FadeRank$_\mathbf{T}$ (resp. FadeRank$_\mathbf{P}$) with the ones obtained by TURank and Forget$_\mathbf{T}$ (resp. PageRank and Forget$_\mathbf{P}$). To compare two rankings we use the Kendall's τ [12], similarly to [27]. The Kendall's τ is a value in the range $[-1; +1]$ which measures the correlation between two rankings: in particular, $\tau = +1$ indicates that the two rankings perfectly agree (i.e. they are the same), while $\tau = -1$ indicates perfect disagreement (i.e., a ranking is the inverse of the other), and $\tau = 0$ denotes no correlation.

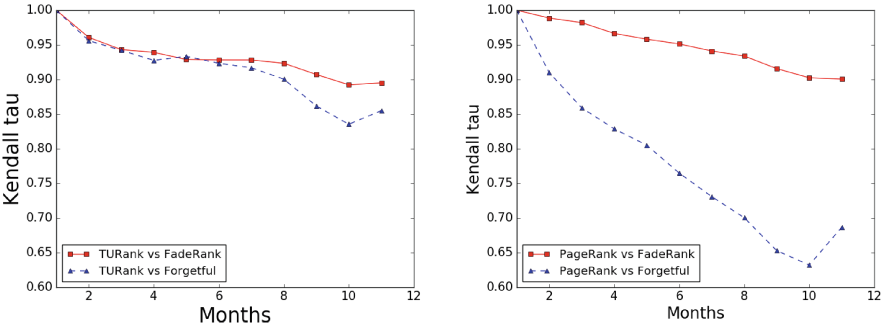
[2] Note that we cannot compute the γ-weights by solving the linear system, because the value D_t^n is a linear combination of the other two (i.e., $D_t^n = R_t^n - H_t^n$).



(a) (b)

Fig. 4. In 4a, the execution time of FadeRank$_{\mathbf{T}}$ (dashed line) and TURank (solid line) for dataset D1. In 4b, the same for FadeRank$_{\mathbf{P}}$ and PageRank.

Figure 5 shows the results of our experiments on dataset D1. We see that, despite its constant-time execution, FadeRank$_{\mathbf{T}}$ loses a small amount of precision with respect to TURank, but it is still more precise than Forget$_{\mathbf{T}}$ on the long term. The comparison of FadeRank$_{\mathbf{P}}$ and PageRank shows similar results: in this case, the gain of precision of FadeRank with respect to the forgetful algorithm is evident from the starting time intervals.



(a) (b)

Fig. 5. In 5, the Kendall's τ correlation between the rankings of FadeRank$_{\mathbf{T}}$ *vs.* TURank (solid line), and Forget$_{\mathbf{T}}$ *vs.* TURank, on dataset D1. In 5b, the same for FadeRank$_{\mathbf{P}}$, Forget$_{\mathbf{P}}$ and Pagerank.

To further compare the precision of FadeRank with that of the forgetful algorithms, we have experimented on an artificial dataset which represents a whitewashing attack scenario. The dataset, containing 500 users and 400 K tweets spanning over 11 months, contains 30 % of spammers, who—after being ranked low in the past—begin to share high-quality content in the attempt to whitewash their reputation. Since TURank and FadeRank process the whole past behaviour of users, they can address this attacks, by letting the reputation of spammers increase slowly. Conversely, the forgetful algorithms discard the past history every month, hence they are susceptible to such whitewashing attacks.

The results of our experiments, reported in Fig. 6, show that the hybrid approach adopted by FadeRank, which only records a bounded digest of the past history (namely, $b^m - 1 = 7$ fading memories), is enough to address the whitewashing attack. More precisely, the diagram in Fig. 6a shows the expected drop of precision for Forget$_{\mathbf{T}}$ with respect to TURank starting at the seventh month, while the precision of FadeRank remains within $\tau > 0.9$.

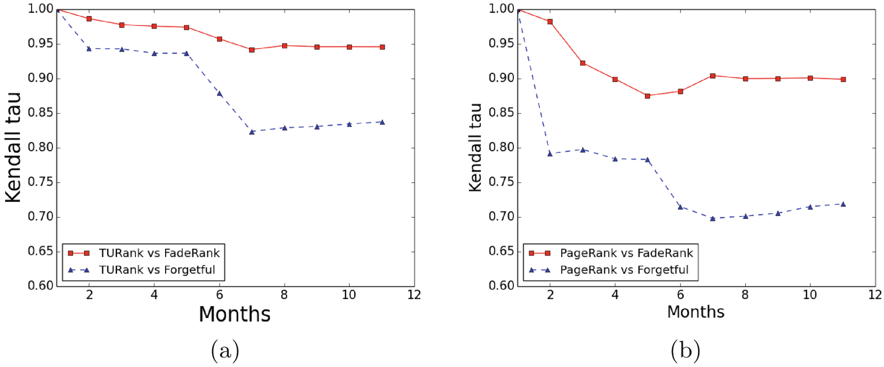


Fig. 6. In 6a, the precision of FadeRank$_{\mathbf{T}}$ and Forget$_{\mathbf{T}}$ for the whitewashing dataset. In 6b, the same for FadeRank$_{\mathbf{P}}$ and Forget$_{\mathbf{P}}$.

5 Conclusions

We have proposed an incremental reputation algorithm for Twitter, which updates the user reputation in (roughly) constant time. In this way we address a performance issue of reputation algorithms, which typically either consider the whole historical data at every update (so suffering from a huge computational overhead), or just discard the past (so being subject to whitewashing attacks).

Our algorithm, named FadeRank, exploits the technique introduced in [23] to summarize the past history in a bounded number of values—the *fading memories*. FadeRank combines the fading memories with the raw reputation computed in the most recent time interval, obtained (in constant time) by an arbitrary reputation algorithm. The actual behaviour of FadeRank depends on the weights associated to these components. A dominant weight on the raw component makes the reputation adapt very quickly to the most recent behaviour: a consequence of this choice is that spammers, i.e. malicious users that frequently tweet uninteresting or misleading content, could adopt a *strategic oscillation behaviour*, leading to a whitewashing attack [10]. A user who adopts this behaviour oscillates between building reputation, behaving non-maliciously, and then “milking” reputation, behaving maliciously. Instead, a dominant weight on the historical component makes the reputation adapt more slowly to the recent behaviour, so offering a defence against the above-mentioned attack.

We have validated FadeRank by comparing its performance and precision with those of two standard ranking algorithms, i.e. TURank [28] and a variant of PageRank [20] tailored to rank Twitter users. To perform the validation we have developed a crawler, through which we have downloaded from Twitter three large datasets of tweet/retweet/follow data, spanning over a period of 11 months. Our experiments show that, although the execution time of FadeRank is nearly constant at each reputation update (while, as expected, the execution time of TURank and PageRank grows linearly at each update), the loss in precision with respect to the raw algorithms is very limited. Further, FadeRank outperforms in precision the *forgetful* versions of the raw reputation algorithms, which naïvely truncate the past history. In particular, our experiments show that these algorithms suffer from a huge loss of precision in the presence of whitewashing attacks, while the ranking obtained by FadeRank is still quite close to that of the raw algorithms.

Acknowledgments. This work has been partially supported by Aut. Reg. of Sardinia P.I.A. 2013 "NOMAD", and by EU COST Action IC1201 "Behavioural Types for Reliable Large-Scale Software Systems" (BETTY).

References

1. Lithium Technologies Acquires Klout. http://www.lithium.com/company/news-room/press-releases/2014/lithium-technologies-acquires-klout. Accessed 09 April 2016
2. Twitter Usage Statistics. Internetlivestats.com http://www.internetlivestats.com/twitter-statistics. Accessed 09 April 2016
3. The top 20 valuable Facebook statistics, December 2015. Zephoria.com https://zephoria.com/top-15-valuable-facebook-statistics. Accessed 09 April 2016
4. Balmin, A., Hristidis, V., Papakonstantinou, Y.: Objectrank: authority-based keyword search in databases. In: VLDB, vol. 30, pp. 564–575 (2004)
5. Bartoletti, M., Cimoli, T., Murgia, M., Podda, A.S., Pompianu, L.: A contract-oriented middleware. In: Braga, C., Ölveczky, P.C. (eds.) FACS 2015. LNCS, vol. 9539, pp. 86–104. Springer, Heidelberg (2016). doi:10.1007/978-3-319-28934-2_5
6. Dimitriou, T., Karame, G., Christou, I.T.: SuperTrust: a secure and efficient framework for handling trust in super-peer networks. In: ACM PODC, pp. 374–375 (2007). http://doi.acm.org/10.1145/1281100.1281180
7. Haveliwala, T.H.: Topic-sensitive pagerank. In: WWW, pp. 517–526. ACM (2002)
8. Hoffman, K.J., Zage, D., Nita-Rotaru, C.: A survey of attack and defense techniques for reputation systems. ACM Comput. Surv. **42**(1), 1 (2009)
9. Hu, W., Zou, H., Gong, Z.: Temporal pagerank on social networks. In: Wang, J., Cellary, W., Wang, D., Wang, H., Chen, S.-C., Li, T., Zhang, Y. (eds.) WISE 2015. LNCS, vol. 9418, pp. 262–276. Springer, Heidelberg (2015). doi:10.1007/978-3-319-26190-4_18
10. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decis. Support Syst. **43**(2), 618–644 (2007)

11. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: ACM SIGKDD, pp. 137–146. ACM (2003)
12. Kendall, M.G.: A new measure of rank correlation. Biometrika **30**(1/2), 81–93 (1938)
13. Lampe, C., Johnston, E.W., Resnick, P.: Follow the reader: filtering comments on slashdot. In: CHI, pp. 1253–1262 (2007)
14. Lampe, C., Resnick, P.: Slash(dot) and burn: distributed moderation in a large online conversation space. In: CHI, pp. 543–550 (2004)
15. Ma, H., Qian, W., Xia, F., He, X., Xu, J., Zhou, A.: Towards modeling popularity of microblogs. Front. Comput. Sci. **7**(2), 171–184 (2013)
16. Mariani, M.S., Medo, M., Zhang, Y.C.: Ranking nodes in growing networks: when pagerank fails. Scientific reports 5 (2015)
17. Michiardi, P., Molva, R.: Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In: Communications and Multimedia Security, IFIP Conference Proceedings, vol. 228, pp. 107–121. Kluwer (2002)
18. Mislove, A., Gummadi, K.P., Druschel, P.: Exploiting social networks for internet search. In: HotNets, p. 79 (2006)
19. Motwani, R., Raghavan, P.: Randomized Algorithms. Chapman & Hall/CRC, London (2010)
20. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. Technical report 1999-66, Stanford InfoLab (1999)
21. Rao, A., Spasojevic, N., Li, Z., DSouza, T.: Klout score: measuring influence across multiple social networks. In: Big Data, pp. 2282–2289. IEEE (2015)
22. Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E.: Reputation systems. Commun. ACM **43**(12), 45–48 (2000)
23. Srivatsa, M., Xiong, L., Liu, L.: TrustGuard: countering vulnerabilities in reputation management for decentralized overlay networks. In: WWW, pp. 422–431. ACM (2005)
24. Uysal, I., Croft, W.B.: User oriented tweet ranking: a filtering approach to microblogs. In: ACM CIKM, pp. 2261–2264. ACM (2011)
25. Vosecky, J., Leung, K.W.-T., Ng, W.: Searching for quality microblog posts: filtering and ranking based on content analysis and implicit links. In: Lee, S., Peng, Z., Zhou, X., Moon, Y.-S., Unland, R., Yoo, J. (eds.) DASFAA 2012, Part I. LNCS, vol. 7238, pp. 397–413. Springer, Heidelberg (2012)
26. Wang, A.H.: Don't follow me: spam detection in Twitter. In: SECRYPT, pp. 1–10. IEEE (2010)
27. Weng, J., Lim, E.P., Jiang, J., He, Q.: TwitterRank: finding topic-sensitive influential twitterers. In: ACM WSDM, pp. 261–270. ACM (2010)
28. Yamaguchi, Y., Takahashi, T., Amagasa, T., Kitagawa, H.: TURank: Twitter user ranking based on user-tweet graph analysis. In: Chen, L., Triantafillou, P., Suel, T. (eds.) WISE 2010. LNCS, vol. 6488, pp. 240–253. Springer, Heidelberg (2010)
29. Yu, H., Gibbons, P.B., Kaminsky, M., Xiao, F.: SybilLimit: a near-optimal social network defense against sybil attacks. In: IEEE S&P, pp. 3–17 (2008)
30. Yu, H., Kaminsky, M., Gibbons, P.B., Flaxman, A.D.: SybilGuard: defending against sybil attacks via social networks. IEEE/ACM Trans. Netw. **16**(3), 576–589 (2008)

Personalized Re-ranking of Tweets

Yukun Zhao[1], Shangsong Liang[2], and Jun Ma[1](✉)

[1] Shandong University, Jinan, China
yukunzhao.sdu@gmail.com, majun@sdu.edu.cn
[2] University College London, London, UK
shangsong.liang@ucl.ac.uk

Abstract. In microblogs, the problem of information overload has troubled many users especially those with numerous followees. Users receive hundreds of tweets in chronological order and have to scan through pages of tweets to find useful information. In this paper, we propose a personalized tweet re-ranking framework for re-ranking the tweets received by a user based on his preference such that interesting tweets are ranked higher for the user. With the personalized re-ranked tweets, the user can find his interesting tweets conveniently. Modeling users' preference in the context of tweet streams is more challenging than modeling that in the context of long documents as it is difficult to capture users' interests with sparse short text documents like tweets. To address this challenge, we propose a media awareness tweet re-ranking model, MATR for short, to incorporate WeMedia accounts (WeMedia is a type of accounts in microblogs that only has media attributes publishing original and valuable messages), and explicitly calculate the influence of the publishers of these tweets. Experimental results demonstrate the effectiveness of our method compared to state-of-the-art baselines.

Keywords: Personalized tweet re-ranking · Topic modeling · Microblog

1 Introduction

With the rising of social media, microblogs such as Twitter[1] and Sina Weibo[2] have become increasingly popular for their important roles in information sharing and interpersonal communication. When a user logs in with his own account, there would be a large amount of tweets shown to the user especially when he has many followees. Almost half of the tweets pushed to him are pointless babble while the rest of them are news, conversations, self-promotions and trashes [8,9,13,16,18]. Some important and useful tweets would be flooded by other tweets that the users do not care. This problem of information overload problem troubles the users, since all tweets are posted to them in chronological order and considered equally important regardless of the users' personalized interests.

[1] http://twitter.com.
[2] http://weibo.com.

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 70–84, 2016.
DOI: 10.1007/978-3-319-48743-4_6

One effective solution to assist a user to access the tweets he is interested in is to re-rank the tweets posted to him based on his personal interests. Users' interests modeling has been widely studied including [4,7,15,19,22,24,26,28], who model users' interests in terms of topics. Tweet ranking and recommendation [2,5,11,25,30], integrate tweet contents and other features including tweet history and social relations to infer users' preferences. All of these methods do not consider media attributes of microblogs and big data behind the users' followees.

To effectively retrieve interesting tweets for a specific user, we propose a personalized media awareness tweet re-ranking model, abbreviated as MATR. We estimate a user's personalized preference based on the tweets he posted and the WeMedia accounts he is following. Our approach builds on the previous work [7,17,26–28,30], but we explicitly consider the big data behind the users' followed WeMedia accounts and the influence of tweet publishers. We briefly describe our model in the following two paragraphs.

Users' media attributes have been discussed in [8,10,12,14,20,29]. WeMedia [20,29] is a type of accounts which focuses on vertical specialization areas such as technology, finance, automobile etc. These WeMedia accounts do publish lots of original and valuable contents [20]. In general, WeMedia accounts publish a large amount of tweets and the texts obey common text rules with more meaningful nouns, phrases and universal grammars. Intuitively speaking, the reason why a user follows a WeMedia accounts is that there are some topics the user concerns in the contents he published. We then incorporate the tweets' texts published by the WeMedia accounts the user is following and the user's published tweets to infer a global topic mode (GTM), via which we can infer the user's topic distribution.

Words can have different meanings in different contexts. For example, if a user often tweets about IT, iphone, coding, android, we need the word "apple" to indicate a product of Apple Inc. instead of fruit when he says "Apple is amazing". In order to solve this problem of word ambiguity, for each document (tweet), we use all the tweets published by the WeMedia account who is the original publisher of the tweet, to build a local topic model (LTM). We use two hierarchical topic models (GTM and LTM) to infer the topic distributions of each tweet and the user's interests. We calculate the user's preference score on a tweet based on the topical similarity between the user and the tweet. And, we calculate the influence of the tweet publisher, including the authority of the publisher, the quality of tweets and social interactions between the user and the publisher. Experimental results show that our approach captures each user's interests more accurately and recalls more useful tweets.

The main contributions of this paper are as follows:

(1) We propose a personalized media awareness tweet re-ranking model (MATR) to re-rank the tweets posted to a user based on the his preference for these tweets.
(2) We alleviate sparsity of data, topic coarse in modeling users' interests by incorporating those large amount of WeMedia accounts the user is following.

(3) We propose to use the local topic models for the tweets to alleviate word ambiguity and understand what the tweets talk about better.
(4) We compute the influence of the publishers of tweets on the user's preference for these tweets explicitly.

The rest of this paper is organized as follows. We review related work in Sect. 2. We give our proposed framework in Sect. 3. Section 4 shows experiments and evaluation. We give conclusions in Sect. 5.

2 Related Work

Our approach builds on the earlier work in automatic WeMedia accounts detection, topic modeling, tweet ranking and tweet recommendation.

2.1 WeMedia

Kwak et al. [8] and Shayne and Willis [29] prove that the microblog service is not only a social network but also a news media platform, while some accounts exist as media accounts and publish many valuable tweets. These accounts who publish vast original and useful messages, are regarded as WeMedia accounts [20]. Liu and Zhang [20] study WeMedia accounts from posting behaviors and posting contents and then propose a method to detect WeMedia accounts automatically, while the method of WeMedia accounts detection is used in our work.

2.2 Topic Modeling

Topic models [1,6] are widely used to project high-dimensional words into low-dimensional latent topics, where each document and each word are viewed as multinomial distributions over a set of topics. The latent topics extracted from users' documents and words are used to infer users' interests [15,19,24,28,35]. When dealing with short texts, Wan and Xiao [31,32] add extra neighbor documents for topic decomposition. But the finding of neighbor documents are usually arbitrary. These methods introduce too much noise and result in topic drift when the document and its so-called neighbor documents do not talk about the same topics. Liu et al. [21] and Matthew and Macskassy [23] aim to find the topics to represent interests for users in Twitter by identifying the entities they mentioned in the tweets. Zhao et al. [4] extract representative key words from tweets with considering the setting of Twitter and classify latent topics into "back-ground" topics and "personal" topics. Inspired by these methods, we represent a user's interests and a tweet in both topic level and word level.

2.3 Tweet Ranking and Recommendation

Weng et al. [33] propose a graph-based ranking strategy to rank tweets posted to each user based on the relevance between users' interests and tweet contents.

Duan et al. [3] use a learning-to-rank approach for general tweet ranking. Feng and Wang [5] incorporates all sources of information like users' profile, tweet quality, interaction history to rank the tweets for each users. Vosecky et al. [30] utilizes topic models and language models to represent words in both topic level and word level, which gives an efficient method to calculate topic affinity between users and tweets. Chen et al. [2] propose to recommend tweets based on collaborative ranking strategy and other useful contextual information. Similarly, recommending "novel" tweets to users are studied in [25]. These methods consider many information including social relations and other explicit features to represent the user's preference for each tweet, but when to infer each user's interests they still decompose the user's tweet contents into topic level using traditional topic models simply without solving the coarse topics and the problem of word ambiguity.

Our work is different from the above related works in the following important ways: (1) We incorporate long tweets published by WeMedia accounts which have meaningful nouns, meaningful phrases and universal grammars to alleviate data sparsity when infer users' interests in the setting of microblogs and (2) we then explicitly calculate the influence of the tweet publisher on the users' preference for a tweet.

3 Media Awareness Tweet Re-ranking

3.1 Overview

We aim to re-rank the tweets that are posted to each user during a certain time period for each user. That is: our task is to estimate each user's preference on each tweet. We use u to represent a user and i to indicate an item (a tweet posted to the user u). A user u's interests are represented as a multinomial distribution of topics M_u, and an item i is represented as a multinomial distribution Q_i. The score $\widehat{r_{u,i}}$ of user u's preference on tweet i is obtained as follows:

$$\widehat{r_{u,i}} = b_i + sim(M_u, Q_i), \qquad (1)$$

Table 1. Main notations used in this paper

Notation	Gloss
K	Number of topics
V	The vocabulary size
u	A user
i	A new posted tweet (item, document)
d	A tweet or retweet
w	A word presenting in a tweet, $w \in d$
M_u	Multinomial distribution of user u
Q_i	Multinomial distribution of item i
b_i	Influence of publisher of tweet i
$\widehat{r_{u,i}}$	Estimation of u's preference on item i
N_u	The WeMedia accounts user u followed
θ_d	Multinomial distribution of tweet d
ϕ_k	Multinomial distribution of topic k
z_w, z_d	Topic assignment on word w and tweet d
α, β	Hyper parameters in our topic model
μ, λ, η	Hyper parameters in MATR

where $sim(M_u, Q_i)$ represents the similarity between a user's topic distribution and the distribution of item i. b_i is the influence of the publisher of tweet i. The main notations we use in this paper is summarized in the Table 1.

We show the method to calculate topical similarity between a user and an item in Subsect. 3.2. Then we detail the influence of the publisher of tweet i on the user u's preference for tweet i in Sect. 3.3.

3.2 Topical Similarity Between a User and an Item

We split the user u's set of published tweets into two subsets. The first subset D_u contains all the user's original tweets, while the second subset R_u contains tweets that are retweeted from other users. In order to represent the user's interests M_u, we need to infer topic distribution M_{D_u} and topic distribution M_{R_u}. They relationship of these three multinomial distributions is the following:

$$M_u = \lambda M_{D_u} + (1 - \lambda) M_{R_u}, \tag{2}$$

where λ is free parameter that measures the importance of original tweets in inferring user u's interests. We use Jelinek Mercer smoothing method [34] to incorporate topic distributions of original tweets and retweets.

Topic Distribution of Original Tweets $\boldsymbol{M_{D_u}}$: We use LDA [1] to infer a topic distribution of each document, in our case, tweet, and then average all distributions to infer M_{D_u}. We use a tweet set D_u published by the user's followees and himself to build a global topic model abbreviated as GTM. We use GTM to infer the topic distribution on tweet d as θ_d, $\theta_{d,k}$ representing the k-th dimension of θ_{t_j}, $d \in D_u, k \in [1, K]$. w is a word in tweet d. ϕ_k^{GTM} is a topic-word distribution being the probability of a set of words generated under topic k. $P(w|\phi_k^{\mathrm{GTM}})$ indicates the probability of word w being generated under topic k. $\theta_{d,k}^{GTM}$ represents the probability of a document d being assigned to topic k in GTM. The user's original tweets topic distribution M_{u,D_u} on topic k can be formulated:

$$M_{D_u,k} = \frac{1}{|D_u|} \sum_{j=1}^{|D_u|} \theta_{d_j,k}^{\mathrm{GTM}}. \tag{3}$$

Then we introduce how to incorporate the WeMedia accounts the user is following to infer topic distributions of retweets.

Topic Distribution of Retweets $\boldsymbol{M_{R_u}}$: For each user u, we identify the WeMedia accounts user u followed as N_u using the method [20]. Inspired by [30], we use both words and topics to represent the user's interests. For each tweet $d \in R_u$, there is a publisher $N_{u_d} \in N_u, N_{u_d} \neq u$. We use the tweets published by the WeMedia account N_{u_d} to train a local topic model LTM^d. Then we get document-topic distribution $\theta_d^{\mathrm{LTM}^d}$ for document d and topic-word distribution $\phi_k^{\mathrm{LTM}^d}$ for each topic k.

We assign the document d a single topic by choosing the topic that maximizes the probability of $\theta_{d,k}$. Then the topic assignment on tweet d is z_d:

$$z_d = \arg\max_k \theta_{d,k}^{\mathrm{LTM}^d} = \arg\max_k \prod_{w \in d} P(w|\phi_k^{\mathrm{LTM}^d}). \tag{4}$$

Similarly, one single topic assignment z_w on word $w \in d$ is obtained by choosing the topic that maximizes the probability of $\phi_{k,w}^{\mathrm{LTM}^d}$:

$$z_w^{\mathrm{LTM}^d} = \arg\max_k P(w|\phi_k^{\mathrm{LTM}^d}). \tag{5}$$

For each word $w \in d$, select top-N words in the topics whose topic is $z = z_w^{\mathrm{LTM}^d}$ and add these words into document d, leading the document d has more precise and sufficient words to express its topics. Then we get document d expanded to $d^{'}$, and the retweet set R_u expanded to $R_u^{'}$. At the same time, users have similar interests with the publisher as mentioned below. After incorporate the WeMedia contents, we formulate the topic distributions of retweet set R_d on topic k as follows:

$$M_{R_u,k} = \frac{1}{|R_u^{'}|} \sum_{d^{'}=1}^{|R_u^{'}|} \theta_{d^{'},k}^{\mathrm{GTM}}. \tag{6}$$

After the two steps inferring M_{D_u} and M_{R_u}, we infer the user u's interests M_u by combining topic distributions of his original tweets M_{D_u} and topic distributions of all retweets M_{R_u}. The k-th dimension of M_u is formulated as:

$$\begin{aligned} M_{u,k} =& \lambda M_{D_u,k} + (1-\lambda) M_{R_u,k} \\ =& \lambda \frac{1}{|D_u|} \sum_{d=1}^{|D_u|} \theta_{d,k}^{\mathrm{GTM}} + (1-\lambda) \frac{1}{|R_u^{'}|} \sum_{d^{'}=1}^{|R_u^{'}|} \theta_{d^{'},k}^{\mathrm{GTM}} . \end{aligned} \tag{7}$$

Topical Representation of an Item $\boldsymbol{Q_i}$: If the publisher of tweet(item) i is an ordinary user, we use GTM to obtain its topic distribution Q_i. The k-th dimension of Q_i is obtained as follows:

$$Q_{i,k} = \frac{\prod_{w \in i} P(w|\phi_k^{\mathrm{GTM}})}{\prod_{w \in i} \sum_z P(w|\phi_z^{\mathrm{GTM}})}. \tag{8}$$

If the publisher of tweet i is a WeMedia account, we get the tweet i expanded to $i^{'}$. We add similaring words under the identical topics in LTM^i as metioned before. Then, we get the topic distribution on tweet i Q_i, while the dimension k of Q_i is obtained as follows:

$$Q_{i,k} = \frac{\prod_{w \in i'} P(w|\phi_k^{\mathrm{GTM}})}{\prod_{w \in i'} \sum_z P(w|\phi_z^{\mathrm{GTM}})}. \tag{9}$$

As we represent each user and each item via a multinomial topic distribution, we calculate the cosine similarity between each user and each item. Then the remaining problem is to calculate the influence of the publisher, which is shown in the next Subsect. 3.3.

3.3 Calculation of Publisher Influence

We formulate the publisher's influence on the user u's preference on item i as b_i. The publisher's explicit features are deduced in the following three primary parts:

Interest Affinity Weight $\boldsymbol{w_I(N_{u_i}, u)}$: We use N_{u_i} to indicate the publisher of tweet i, and $w_I(N_{u_i}, u)$ to indicate the affinity of interests between the user u and the user N_{u_i}. Generally, the user tend to retweet a tweet when the tweet publisher has similar interests with him. We measure the similarity of interests between the user u and the publisher N_{u_i} via their latent topic profiles using inverse KL-divergence:

$$w_I(N_{u_i}, u) = 1/KL(M_u, M_{N_{u_i}}). \tag{10}$$

Publisher Authority Weight $\boldsymbol{w_A(N_{u_i}, u)}$: The authority of a publisher is indicated by the number of followers, the number of tweets and the number of mentioned times. We use c_{follow} to indicate the number of followers of user N_{u_i}, and use $c_{followAvg}$, $c_{followMax}$ indicate the average number of followers a user is following and the maximal number of followers a user is following respectively. The number of tweets infer the user's activeness in microblogs. c_{tweet} is number of tweets published by user N_{u_i}, $c_{tweetAvg}, c_{tweetMax}$ are the average number of tweets a user published and the maximum number of tweets a user published respectively. Mentioned counts are the times the publisher has been mentioned in other tweets, the frequency indicating the popularity of the publisher. c_{ment}, $c_{mentAvg}$ and $c_{mentMax}$ represent the number of times user N_{u_i} being mentioned in other tweets, the average number of times a user being mentioned and the maximum number of times a user being mentioned. The authority weight is normalized as follows:

$$\begin{aligned} w_A(N_{u_i}, u) =& \eta_1 \frac{c_{follow} - c_{followAvg}}{c_{followMax}} + \eta_2 \frac{c_{tweet} - c_{tweetAvg}}{c_{tweetMax}} + \\ & \eta_3 \frac{c_{ment} - c_{mentAvg}}{c_{mentMax}}. \end{aligned} \tag{11}$$

Content Quality Weight $\boldsymbol{w_C(N_{u_i}, u)}$: The quality of a tweet is estimated by the length of the tweet, retweeted times of the tweet, the number of comments in the tweet. Tweets which are long and retweeted or commented many times could be awarded as high quality tweets. We use $c(awardedTweet)$ to indicate the number of high quality tweets and use $(c(tweet)$ to indicate the number of tweets the user N_{u_i} published. The weight of the quality of publisher's contents is estimated by $w_C(N_{u_i}, u)$:

$$w_C(N_{u_i}, u) = \frac{\log(c(awardedTweet))}{\log(c(tweet))}. \tag{12}$$

The overall weight of the influence of publisher N_{u_i} on user u is formulated as:

$$b_i = \mu_1 w_I(N_{u_i}, u) + \mu_2 w_A(N_{u_i}, u) + \mu_3 w_C(N_{u_i}, u), \tag{13}$$

where $\mu_1, \mu_2, \mu_3 \in [0, 1]$ indicates the weight of different factors, $\sum_j \mu_j = 1$.

Using Eqs. 3, 7, 8, 9 and 13, the user $u's$ preference score $\widehat{r_{u,i}}$ on item i is formulated as follows:

$$\widehat{r_{u,i}} = \cos(M_u, Q_i) + \mu_1 w_I(N_{u_i}, u) + \mu_2 w_A(N_{u_i}, u) + \mu_3 w_C(N_{u_i}, u). \tag{14}$$

4 Experiments

4.1 Experimental Setup

In this paper, we use Sina Weibo as our default setting of microblog service. We work with a dataset crawled from Sina Weibo. The dataset contains 896 users including all their published tweets from the date of their registration up to May 1, 2015 and their social relations. For each user, we crawl all the followees he followed including the tweets and user-information. Finally, we get 21058 users in total. The average number of tweets an ordinary user published is 1,137, including 156 original tweets and 981 retweets. In these 981 retweets, 563 of them are published from WeMedia accounts while the rest of them are published from ordinary users. An ordinary user follows 133 users averagely, and 64 of them are WeMedia accounts. Table 2 shows the statistics of the dataset.

Table 3 shows the difference between ordinary users and WeMedia accounts. We get 4864 WeMedia accounts and 16194 ordinary users in this dataset. Averagely, a WeMedia account has 12798 followers while an ordinary user only has 433 followers. The average number of tweets and average retweeted times per tweet published by a WeMedia account are 9799 and 97 respectively, while the values are 1137 and 0.86 respectively published by an ordinary user. The average length of a original tweet published by a WeMedia account is 76 while the value is

Table 2. For each user, #tweets to be trained, #followees, #WeMedia accounts he followed, #tweets, #retweets

Training tweets	Followees	WeMedia accounts	Original tweets	Retweets
2.3M	133	64	156	981

Table 3. Comparison between WeMedia accounts and ordinary users, #, #followers, #tweets(including original tweets and retweets), retweeted times per tweet, average length of each tweet

Account type	#	#followers	#tweets	Retweeted times	Tweet length
WeMedia accounts	4864	12798	9799	97	76
Ordinary users	16194	433	1137	0.86	23

only 23 when published by a ordinary user. The numbers of key words of a tweet published by ordinary users and WeMedia accounts are shown in Subsect. 4.5.

In this dataset, the retweeted tweets are regarded as positive samples and the others are negative samples. For each user, we depart the tweets in two parts based on their chronological order, and the first three fourths of the tweets are used for training while the rest of them are used for validation.

4.2 Effectiveness of MATR

We use precision to indicate the ratio of tweets in the ranked list that are retweeted finally, and use recall to indicate how many tweets the user has retweeted can be found by our method. We use Mean-F1 measure to evaluate our method. Mean-F1 is obtained by averaging F1 values of the all 896 users.

Now we show the performance of our method compared to other four methods. The detailed implementations are listed below:

Chronological: The tweets are regarded as equally important and posted in chronological order. This strategy indicates the default user experience in microblogs.

Retweeted Times: We re-rank the tweets based on its retweeted times, for the retweeted times is an objective estimation of the popularity of a tweet. This method ignores personalized users' interests and regards all users' interests are the same as the public.

LDA: In LDA [1], user interests are represented from the majority of tweets he published. Known the distribution of user u and tweet d, we calculate the preference score as below:

$$y_{u,d} = \sum_{d_0 \in Tweets(u)} D_{KL}(d_0||d) + b_d \tag{15}$$

Here $D_{KL}(d_0||d)$ express the KL-divergence between the topic distribution of two tweets, b_d express tweet bias.

CTR: Collaborative tweet recommendation [2], a representative method in the state-of-art, which incorporates users' contents and social relations. On the one hand, CTR decompose users' tweet contents into topic level using traditional latent topic models to capture user personal interests. On the other hand, CTR take social relation factors and explicit features into account to represent each users' preference to each particular tweet.

Figure 1 shows the performance of five compared methods. In general, the re-ranked tweet lists should be compact, then the length of re-ranked tweet lists are 10, 30, 50 and 100 in our experiments.

In Fig. 1, we see that chronological strategy shows poor results with Mean-F1 value 0.095 when the length of a list is only ten, which indicates users can almost retweet no more than one status in the ranking list. When the length of ranking list increase to 30, 50 and 100, its Mean-F1 value is 0.1, 0.1 and 0.12. It is obvious that users scan the tweets but do not retweet frequently. This method in our experiments is regarded as a random order to present statuses to a personalized user and the performance is depended on the ratio of positive samples.

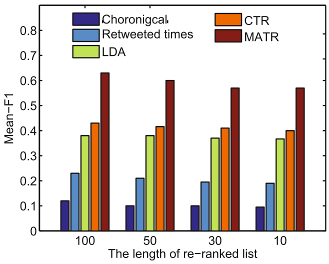


Fig. 1. Mean-F1 of five compared methods in different length of tweet lists.

The performance based on retweeted times obtains a slight improvement compared with chronological order, whose Mean-F1 values range from 0.19 to 0.23. It is still a poor result for there are many positive samples in test dataset indicating that even if we rank the tweets randomly we can obtain a comparable performance with retweeted times strategy. The result verify the necessity of personalized tweet re-ranking while personalized users' interests are not very similar to the popular interests.

LDA outperforms the two mentioned methods because it use users' tweets to infer their interests, and the Mean-F1 values are 0.367 to 0.38. When the length of tweet list varies from 10 to 30, the Mean-F1 values vary from 0.367 to 0.37. The Mean-F1 value slightly improves to 0.38 with the length of re-ranked item list increased to 50 and 100.

Then we come to CTR, the collaborative tweet ranking method obtain a large improvement compared with three previous methods, whose Mean-F1 values are 0.40, 0.41, 0.416 and 0.43 and the corresponding length the lists are 10, 30, 50 and 100. Our method MATR is built on the basis of CTR, whose Mean-F1 value is improved to 0.63 when the list length is 100, which means we can capture users' preference on the tweets more accurately and precisely with just recommending the top 100 tweets. We see Mean-F1 value will be 0.60, 0.57 and 0.57 when the item list length decreases to 50, 30 and 10.

In the setting of Sina Weibo, a page is composed of three subpages within 45 tweets but mostly users only scan the previous 2 subpages meaning that they only scan top 30 tweets. We select top 30 items from the ranked item list and show the Mean-F1 value and recall value in Table 4. Mean-F1 values of chronological order method, retweeted times, LDA, CTR and MATR are 0.1, 0.195, 0.37, 0.41 and 0.57 respectively as we analyzed before. Recall values of compared four methods are 0.12, 0.26, 0.42, 0.60 and 0.78 respectively. Users scan pages to pages of tweets in microblogs and they concern more about how many tweets they likes can be recalled, thus the big improvement to recall useful tweets demonstrate the efficiency and practicability of our method.

Table 4. When the length of ranked item list is 30, Mean-F1 and recall of chronological, retweeted times, LDA, CTR and MATR

	Chronological	Retweeted times	LDA	CTR	MATR
Mean-F1	0.1	0.195	0.37	0.41	0.57
Recall	0.12	0.26	0.42	0.60	0.78

4.3 Effectiveness of Components

In the previous subsection, we have validated the effectiveness of our proposed method MATR. The Fig. 2 shows the influence of each component, i.e., publisher influence and WeMedia.

The implementation of LDA for personalized tweet ranking has been discussed above. "MATR - WeMedia" represents we exclude WeMedia accounts to the inference of users' interests but consider the tweet publisher's influence. "MATR - Publisher Influence" represents that we incorporate his followed WeMedia accounts to inter the user's interests, without considering publisher influence. MATR is the complete model we have proposed in this paper.

In this Fig. 2, the x-axis is the length of item list 10, 30, 50 and 100 while the y-axis is recall value. Performance of LDA is shown for comparison. The result shows when integrate publisher influence, the values of recall are 0.625, 0.56, 0.56, 0.55 better than LDA whose values of recall are 0.41, 0.424, 0.42, 0.42. Considering the publisher influence means a tweet would be ranked front if its publisher is a popular person or a friend. When incorporate WeMedia accounts, we see recall is improved to 0.75 compared with LDA in 100 ranked items. This big improvement testify the effectiveness of incorporating WeMedia accounts for personalized tweet ranking. The better ranking performance is due to integrating the vast compact tweet contents, which are published by WeMedia accounts, to infer user interests more accurately. "MTAR - Publisher Influence" outperforms "MTAR - WeMedia", which means WeMedia accounts have bigger contribution on personalized tweet re-ranking than publisher influence. We conclude that both publisher influence and WeMedia are helpful in personalized tweet re-ranking.

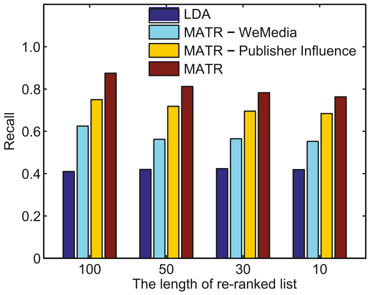


Fig. 2. Comparison of components with different item list length.

4.4 Analysis of Parameters

We trained our model and other baseline topic models using 500 iterations and set $\alpha = 0.5, \beta = 0.1, K = 50$. When we train each LTM, the size of topics is also $K = 50$. Other parameters, i.e., the publisher influence on tweet b_i is calculated explicitly as we mentioned before, and the weights of different factors in publisher influence σ_i are all set 0.33. Figure 3 shows the influence of parameter λ.

λ describes the weight of a user's original published tweets in modeling the user's interests. In the Fig. 3, we see when λ is around 0.3 the recall reaches top. When λ exceeds 0.3, λ increases but recall decreases. This phenomenon shows that larger weight of user's original tweets lead to less ideal ranking results. When λ reaches to 0.9 or more, recall stay close to 0.4 which is close to the performance of LDA method. We conclude that model with smaller λ value means bigger influence from WeMedia accounts leading to a better ranking performance. This result validates the necessity and effectiveness to incorporate WeMedia accounts for personalized tweet re-ranking.

4.5 Quality in Modeling User Interests

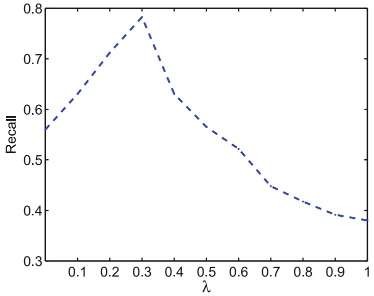
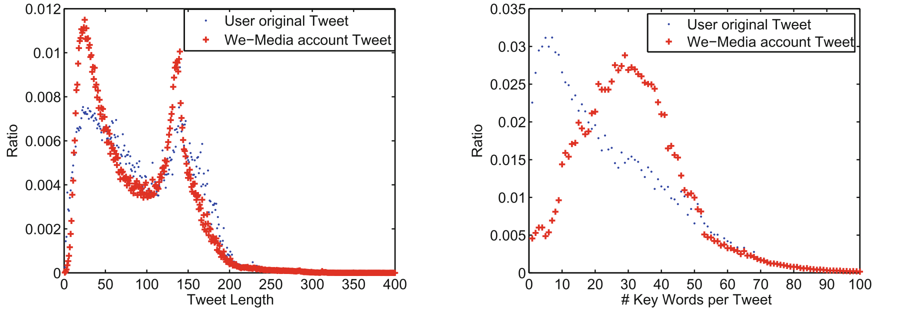


Fig. 3. Influence of λ on recall, with the length of item list being 30.

We use terminology "user original tweet" to denote the tweets originally published by ordinary users (not retweets), "WeMedia account tweet" to denote the tweets originally published by some WeMedia accounts. Figure 4 show average tweet length and average number of key words in a tweet published by ordinary users and WeMedia accounts repectively. As can be seen in Fig. 4(a), the ratio of tweets of which the length is less than 50 words is about 36.5 % from ordinary users, while that is 29.0 % from WeMedia accounts, which indicates that the ratio of tweet length exceeds 50 words is about 71 % published by WeMedia accounts while this ratio is only 63.5 % published by ordinary users. From Fig. 4(b) we see that the number of key words per tweets published by WeMedia tend to be larger than those published by common users. The ratio of number of key words less than 20 is 50.6 % published by ordinary users while the ratio is only 25.5 % published by WeMedia accounts. This phenomenon testify our intuition that tweets published by common users are always short and noisy but tweets published by WeMedia tend to have more compact and expressive words.

Then we shows some words in a tweet represented by topics produced by LTM+GTM and LDA. The tweets are written in chinese originally. For understanding, we translate it into english as follows, 1st tweet: "No matter you are single or married now, please believe that your life will never be lonely when you



(a) Average Tweet Length

(b) Number of Key Words per Tweet

Fig. 4. The ratio of average tweet length and the number of key words per tweet, published by the ordinary users and the WeMedia accounts respectively.

Table 5. The expanded words of original tweets extracted by our method LTM+GTM and LDA, respectively. Words marked blue represent the most coherent words; Words marked green represent less coherent words and others represent irrelevant words.

	Single, married, life									
LTM + GTM	Love	Dream	My dear	Life	Youth	Beijing	Time	Women	Movie	Story
LDA	Dream	Youth	Child	Today	Drama	Friend	Bullshit	Time	Dress up	Brain
	God, good luck									
LTM + GTM	God	Bad luck	Future	Creature	Safeness	Constellation	Legend	Smile	Life	Pisces
LDA	Year	Wine	Child	Variation	Bad luck	God	Creature	Senior	Work	House

get old." After do POS tagging and key phrase extraction we get words "single, married, life" as the central words for this tweet. Then we use LTM to get the words "single", "married", "life" expanded. The top-10 words in the correlate topics is shown in Table 5. The 2nd tweet: "The most cute God, retweet it and you will get good luck." We get the words "God", "good luck" to be trained in our topic model. Each word is assigned a topic and get expanded. We extract top ten words with maximum likelihood in the topics to represent the original tweet in Table 5.

From Table 5, we see that LTM+GTM methods discovers more meaningful and related words for the tweet. And the words in the topic are more coherent. The results demonstrate that we can alleviate word ambiguity and understand what the tweets talk about better. We use consistent topics to get words expanded enhancing their expressibility. In fact, we incorporate long tweets(documents) published by WeMedia accounts instead of directly applying LDA, to alleviate data sparsity, topic coarse in modeling users' interests.

5 Conclusion

In this paper, we propose a novel and efficient framework, MATR, to re-rank the tweets posted to a specific user based on the his preference for the tweets he receives. First, we propose to incorporate each user's followed WeMedia accounts, whose published tweets are always with meaningful nouns, phrases and universal grammars, to infer each user's personal interests. Then, we explicitly calculate the influence of the publishers of tweets including social interactions, the quality of tweet contents and the publishers' authority. Finally, we provide a list of tweets having been re-ranked to the personalized user. Experimental results demonstrate that we capture a user's interests more precisely by better understanding what the tweets are talking about, and our approach recall more useful tweets to the user which is helpful to improve user experience in microblogs.

As future work, we need to improve the efficiency of our proposed algorithm in order to deploy such service on a real social networking platform. Another future work is to focus on the topic extraction on short texts and find the semantic relations in different environments not limited to microblogs.

Acknowledgements. This work is supported by the Natural Science Foundation of China (61672322, 61672324, 61272240, 71402083), the Natural Science foundation of Shandong province (ZR2012FM037), the Excellent Middle-Aged and Youth Scientists of Shandong Province (BS2012DX017) and the Fundamental Research Funds of Shandong University.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
2. Chen, K., Chen, T., Zheng, G., Jin, O., Yao, E., Yu, Y.: Collaborative personalized tweet recommendation. In: SIGIR, pp. 661–670 (2012)
3. Duan, Y., Jiang, L., Qin, T., Zhou, M., Shum, H.-Y.: An empirical study on learning to rank of tweets. In: COLING, pp. 261–270 (2010)
4. Zhao, W.X., et al.: Topical keyphrase extraction from twitter. In: ACL (2011)
5. Feng, W., Wang, J.: Retweet or not? Personalized tweet re-ranking. In: WSDM (2013)
6. Hofmann, T.: Probabilistic latent semantic indexing. In: SIGIR, pp. 50–57 (1999)
7. Krieger, M., Ahn, D.: TweetMotif: exploratory search and topic summarization for twitter. In: ICWSM (2010)
8. Kwak, H., Lee, C., Park, H., Moon, S.: What is twitter, a social network or a news media? In: WWW (2010)
9. Liang, S.: Fusion and diversification in information retrieval. Ph.D. thesis, University of Amsterdam (2014)
10. Liang, S., de Rijke, M.: Finding knowledgeable groups in enterprise corpora. In: SIGIR 2013 (2013)
11. Liang, S., de Rijke, M.: Burst-aware data fusion for microblog search. Inf. Process. Manag. **51**(2), 89–113 (2015)
12. Liang, S., de Rijke, M.: Formal language models for finding groups of experts. Inf. Process. Manag. **52**(4), 529–549 (2016)
13. Liang, S., Rijke, M., Tsagkias, M.: Late data fusion for microblog search. In: Serdyukov, P., Braslavski, P., Kuznetsov, S.O., Kamps, J., Rüger, S., Agichtein, E., Segalovich, I., Yilmaz, E. (eds.) ECIR 2013. LNCS, vol. 7814, pp. 743–746. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36973-5_74
14. Liang, S., Ren, Z., de Rijke, M.: Fusion helps diversification. In: SIGIR, pp. 303–312 (2014)
15. Liang, S., Ren, Z., de Rijke, M.: Personalized search result diversification via structured learning. In: KDD, pp. 751–760 (2014)
16. Liang, S., Ren, Z., Rijke, M.: The impact of semantic document expansion on cluster-based fusion for microblog search. In: Rijke, M., Kenter, T., Vries, A.P., Zhai, C.X., Jong, F., Radinsky, K., Hofmann, K. (eds.) ECIR 2014. LNCS, vol. 8416, pp. 493–499. Springer, Heidelberg (2014). doi:10.1007/978-3-319-06028-6_47
17. Liang, S., Ren, Z., Weerkamp, W., Meij, E., de Rijke, M.: Time-aware rank aggregation for microblog search. In: CIKM, p. 10 (2014)
18. Liang, S., Yilmaz, E., Kanoulas, E.: Dynamic clustering of streaming short documents. In: KDD. ACM (2016)
19. Liang, S., Cai, F., Ren, Z., de Rijke, M.: Efficient structured learning for personalized diversification. IEEE Trans. Knowl. Data Eng. (to appear)
20. Liu, J., Zhang, M.: A study on we media account detection in Sina Weibo. In: CCIR (2014)

21. Liu, X., Zhang, S., Wei, F., Zhou, M.: Recognizing named entities in tweets. In: ACL (2011)
22. Liu, Z., Chen, X., Zheng, Y., Sun, M.: Automatic keyphrase extraction by bridging vocabulary gap. In: CoNLL, pp. 135–144 (2011)
23. Matthew, M., Macskassy, S.A.: Discovering users' topics of interest on twitter: a first look. In: Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data, pp. 73–80 (2010)
24. O'Connor, B., Krieger, M., Ahn, D.: TweetMotif: exploratory search and topic summarization for twitter. In: ICWSM (2010)
25. Pennacchiotti, M., Silvestri, F., Vahabi, H., Venturini, R.: Making your interests follow you on twitter. In: CIKM (2012)
26. Ren, Z., Liang, S., Meij, E., de Rijke, M.: Personalized time-aware tweets summarization. In: SIGIR (2013)
27. Ren, Z., Peetz, M.-H., Liang, S., van Dolen, W., de Rijke, M.: Hierarchical multilabel classification of social text streams. In: SIGIR, pp. 213–222 (2014)
28. Rosen-Zvi, M., Griffiths, T., Steyvers, M., Smyth, P.: The author-topic model for authors and documents. In: UAI, pp. 487–494 (2004)
29. Shayne, B., Willis, C.: We Media: How Audiences are Shaping the Future of News and Information. Media Center at The American Press Institute, Reston (2003)
30. Vosecky, J., Leung, K.W.-T., Ng, W.: Collaborative personalized twitter search with topic-language models. In: SIGIR (2014)
31. Wan, X., Xiao, J.: Collabrank: towards a collaborative approach to single-document keyphrase extraction. In: COLING, pp. 969–976 (2008)
32. Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: AAAI (2008)
33. Weng, J., Lim, E.-P., Jiang, J., He, Q.: Twitterrank: finding topic-sensitive influential twitterers. In: WSDM (2010)
34. Zhai, C.X.: Statistical Language Models for Information Retrieval. Synthesis Lectures on Human Language Technologies. Morgan & Claypool, San Rafael (2008)
35. Zhao, Y., Liang, S., Ren, Z., Ma, J., Yilmaz, E., de Rijke, M.: Explainable user clustering in short text streams. In: SIGIR, pp. 155–164 (2016)

Ranking Microblog Users via URL Biased Posts

Yongjun Ye[1,2], Peng Li[1,2](✉), Rui Li[1,2], Meilin Zhou[1,2], Yifang Wan[1,2], and Bin Wang[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{yeyongjun,lipeng,lirui,zhoumeilin,wanyifang,wangbin}@iie.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China

Abstract. Finding high-quality users to follow is essential for acquiring information in microblogging systems. Measuring user's quality according to its published posts is effective but also needs a large computation considering the volume and the diversity of the posts. In this paper, we explore using only the posts with URLs, i.e., a subset (∼20 %) of the whole posts, for ranking microblog users and propose an iterative graph based ranking algorithm called UBRank to simultaneously rank users and URLs with the assumption that the importance of users and URLs can be mutually boosted. Experiments based on a Chinese microblog corpus demonstrate the effectiveness of the proposed approach.

Keywords: User quality measure · User behavior model · Graph based ranking

1 Introduction

Nowadays, microblogging service has become one of the most important information portals for the Web. It is estimated that 500 million messages[1] are published every day in Twitter.

According to the current system, people have to follow other users (called followees) to keep track of the recent information. The posts of the followees will completely determine the information presented. Identifying high quality users would not only help people to choose followees meeting their interests, but also benefit many applications such as recommendation and information filtering.

Existing works have studied the problem as identifying authority users [1]. However authority does not necessarily means "good" for information seekers. Authority users do provide high quality and credible information, but they may update information very rarely. Those high quality users should also act as information hubs, i.e., people can obtain various and relevant information by following them. The most natural way to measure user quality is to evaluate their posts directly, i.e., representing users by their posts. Previous work [2] has explored to use User-Tweet graph for user ranking, where all of the posts are considered as input. However, from the computation perspective, it may not be that efficient since many posts do not contain valuable information.

[1] http://www.internetlivestats.com/twitter-statistics/.

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 85–93, 2016.
DOI: 10.1007/978-3-319-48743-4_7

In this paper, we explore using only posts with URLs (or posted URLs equally) for measuring user quality. Specifically, our contributions are as follows:

(1) We find that user's posts with URLs are good enough indicators for representing user quality.
(2) We propose to use posted URLs and the related user publish and retweet behaviors for computing user quality. On extracting these behavior data, we only use the posts with URLs as our input instead of all the posts.
(3) We propose a graph based ranking algorithm called UBRank which combines the authority factor and the hub factor together to measure user quality.

The rest of the paper is organized as follows: Sect. 2 introduces related works. Section 3 studies the quality of the posts with URLs and its advantage for representing user quality. Section 4 presents our proposed ranking algorithm. Experiments and evaluation results are provided in Sect. 5. We conclude our paper and discuss possible future work in Sect. 6.

2 Related Work

The most related works are about identifying influential users through using different kinds of information available in Twitter. Measuring user's influence is a well studied problem since the born of microblog [3–6]. Recent works attempt to make a detailed distinction, i.e., identifying topic specific influential users or topical experts [7–9]. In the above studies, influential uses are defined as people with certain authority within its social network [8]. However, none of the above works considered measuring user's importance as an information hub. One notable work is [2]. Specially, [2] construct a user-tweet graph which takes all the tweets into accounts. The number of user's tweets will affect user's quality score. Similar to [2], we evaluate user's quality by considering authority and hub factors simultaneously. The advantage of our work is that we build a more concise graph using only posted URLs, which can significantly accelerate the user evaluation process.

From the perspective of information types, the following relationship [1,2,7,8], publish behavior [2], retweet behavior [1,2,5] and text contents of posts [1,5,8] have all been explored. The above studies all take the user following relationship into consideration for user ranking. Besides, Twitter Lists, which are contributed by micro-blog users, were also explored for finding topical authority [11] and it was found that they can yield more accurate prediction than the systems based on user's bio or tweet content, but the Lists information may not be that common.

From the perspective of methodology, existing studies on ranking users are mostly based on graph ranking algorithms such as PageRank (Page et al. [12]) and HITS (Klienberg et al. [13]). Different structure and iterative algorithm are proposed for different purposes [2,8,10]. Other methodology for finding topical users are based on prediction algorithm, which used many attributes for ranking user quality [1].

3 Analyzing Posts with URLs

3.1 Dataset Construction

We select Sina Weibo as our data source[2] and selected the top 10 seed users with the most followers and who tagged itself with Natural Language Processing (NLP) or Machine Learning (ML). Then expanded the seed users by their following relationship and filtered users by their tags. After filtering, we have got 3,122 users for statistical analysis. In this section, we try to keep as many users for statistical analysis while actually the most related tags are the top 10 tags, which corresponds to 1,073 users. For each user, we crawled the recent 6,000 posts at most. The final dataset contains 5,503,824 posts, for which the publish date range from 2009-08-14 to 2016-01-08.

3.2 Quality Study

The quality of a post is judged on a scale of 0–2 with 0 meaning irrelevant, 1 meaning relevant and 2 meaning "more relevant and interesting" (the criteria is similar to the criteria of NDCG in Information Retrieval Evaluation). The relevance is judged based on the post's topic to the publisher's tags. To label the quality, we sampled users and posts considering the huge volume.

Let Θ_{URL} be the mean average quality score across users for the posts with URLs; $\Theta_{\overline{URL}}$ be the corresponding mean average quality score for the posts without URLs. The null hypothesis is $H_0 : \Theta_{URL} = \Theta_{\overline{URL}}$ and the alternative hypothesis is $H_1 : \Theta_{URL} > \Theta_{\overline{URL}}$. Through manual labeling, we observed that $\Theta_{URL} = 0.93$ while $\Theta_{\overline{URL}} = 0.29$. We conducted the student's t-test and the Wilcoxon signed-rank test (the non-parametric form). Both results show that the null hypothesis is rejected, where $p = 1.6e^{-15}$ and $p = 4.1e^{-11}$ separately. This indicates that posts with URLs have higher quality than posts without URLs.

We further analyzed the average quality score against the number of followers. Figure 1 presents the scatter plot of average post score of each user.

Besides, we made a statistical description on the whole post set. The posts with URLs accounts for 20.8 % of the whole. Figure 2 presents the scatter plot of average retweet count for each user. Obviously, the posts with URLs have a larger retweet counts than the posts without URLs. This means it is easier to evaluate the quality of the posts with URLs than to evaluate the posts without URLs since they have more information available. Another interesting point in Fig. 2 is that the user who has got the largest average retweet count is not the user with the most followers. This shows that the number of user followers cannot fully determine the user's influence, which has been confirmed in [8].

[2] Sina Weibo is one of the most popular microblogging service in China.

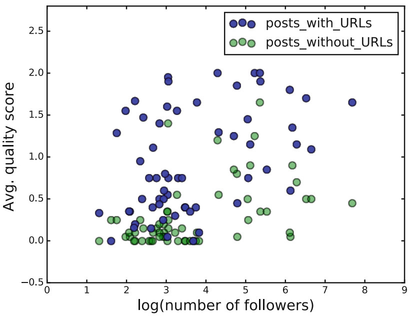


Fig. 1. Avg. quality score for the posts with (without) URLs vs. the number of followers

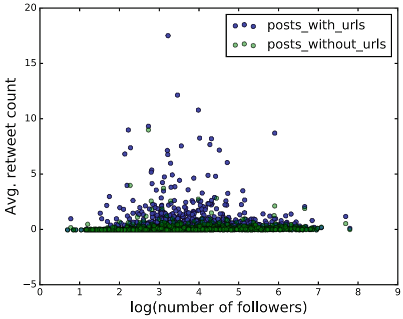


Fig. 2. Avg. retweet counts for the posts with (without) URLs vs. the number of followers

4 URL Biased User Rank (UBRank)

In this section, we describe our approach to calculate user quality score based on the publish behavior and retweet behavior.

4.1 Graph Construction

User-User Graph. Given the user collection $S = \{S_i \mid 1 \leq i \leq m\}$, the User-User graph is a directed graph where each user is considered as a node and a directed edge between two users is formed if one user retweet any post of another user. Specifically, if S_i retweets a post from $S_j (i \neq j)$, then we construct a directed edge from S_i to S_j, donated by $S_i \to S_j$. The weight from S_i to $S_j (i \neq j)$ is set to the number of total retweet counts from S_i to S_j, donated by $\sum S_i \to S_j$. For better formalization, we use an adjacency matrix $\boldsymbol{U}$ to represent weights between user pairs. $\boldsymbol{U} = [U_{ij}]_{m \times m}$ is defined as follows:

$$U_{ij} = \begin{cases} \sum S_i \to S_j, & \text{if i} \neq \text{j} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$\boldsymbol{U}$ can be normalized to $\widetilde{\boldsymbol{U}}$ to make the sum of each row equal to 1:

$$\widetilde{U}_{ij} = \begin{cases} \sum S_i \to S_j \Big/ outdegree(S_i), & \text{if outdegree}(S_i) \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

User-URL Graph. Given the URL collection $T = \{T_i \mid 1 \leq i \leq n\}$ which are extracted from posts, the User-URL graph is an undirected graph where each URL and user is considered as a node. If user S_i posts or retweets a post with URL T_j, donated by $S_i \leftrightarrow T_j$, then we construct an undirected edge from S_i

to T_j. The weight for each edge is set to 1. We use an adjacency matrix $\boldsymbol{V}$ to represent weights for each edge. $\boldsymbol{V} = [V_{ij}]_{m \times n}$ is defined as follows:

$$V_{ij} = \begin{cases} 1, & \text{if } S_i \leftrightarrow T_j \text{ exists} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

The weight from user to URL $\boldsymbol{V}$ is normalized to $\widetilde{\boldsymbol{V}}$. The weight from URL to user, donated by $\boldsymbol{V^T}$, can be normalized to $\widehat{\boldsymbol{V}}$ in the same way.

4.2 Iterative Algorithm

Given the user quality score $\boldsymbol{\upsilon} = [\upsilon(s_i)]_{m \times 1}$ and the URL quality score $\boldsymbol{\nu} = [\nu(t_j)]_{n \times 1}$. The score of user is determined by the scores of its neighbor users and its posts. The corresponding matrix form is:

$$\boldsymbol{\upsilon} = \alpha \widetilde{\boldsymbol{U}}^{\boldsymbol{T}} \boldsymbol{\upsilon} + \beta \widehat{\boldsymbol{V}}^{\boldsymbol{T}} \boldsymbol{\nu}, \quad \boldsymbol{\nu} = \widetilde{\boldsymbol{V}}^{\boldsymbol{T}} \boldsymbol{\upsilon} \tag{4}$$

where α and β determine the importance of quality scores contributed from the homogeneous nodes and the heterogeneous nodes respectively. $\alpha + \beta = 1$.

On constructing graph, the main differences between our work and TuRank [2] are as follows: (1) We model retweet action as user relation instead of post relation and we consider the join effect accumulated from each retweet instead of using retweet to measure each post quality. (2) We do not take user following relationship into the graph though they can be easily incorporated.

5 Experimental Evaluation

5.1 Methods for Comparison

To validate the effects of our proposed approach, we implemented the following methods for user ranking:

UBRank: As described in Sect. 4, UBRank focuses on the posts with URLs and is based on the User-User Graph and the User-URL Graph. The parameter α and β are both set to 0.5, as obtained by training.

RTRankU: This method constructs the User-User graph based on the retweet information of the posts with URLs and ignores the User-URL Graph.

RTRankA: This method constructs the User-User graph based on the retweet information of all the posts including the posts without URLs and ignores the User-URL Graph.

TuRank: TuRank takes following behavior, publish behavior and retweet behavior into consideration. Especially, the constructed graph contains all the post nodes and the reweet action is expressed as the relationship between posts. The edge weights for different relations are set to the values in [2].

TwitterRank: This model is a simplified version of work [8]. We skip the process of calculating user topics from posts, since our users are already picked up within a certain topic. This method constructs a User-User graph based on the following relationship.

Besides, we also implemented some heuristic methods for comparison.

Follower_Count: The users are ranked based on the number of followers.
URL_Count: The Users are ranked based on the number of its posted URLs.
URL_RT_Count: The users are ranked based on the number of total retweet counts of all the user's posts with URLs.
ALL_RT_Count: The users are ranked based on the number of total retweet counts of all the user's posts whether the post has URLs or not.

5.2 Evaluation Results

In this paper we take a similar paradigm for evaluation: we only consider the top ranked users of each method and manually label their quality. Specifically, top 10 users were selected for pooling as [1]. After the pooling, we have 46 unique users for labeling. To be fair for the compared methods, we use all the posts with or without URLs to evaluate user's quality.

Considering the large amount, we use stratified sampling to sample posts and then label their quality. This sampling paradigm is similar to statMAP procedure for IR evaluation. To measure the performance of each method, we use Kendall's τ as our evaluation metric as [8]. A large τ value means the rank is closer to the human judgment.

Effects of UBRank: The evaluation results are given in Table 1. Obviously, the proposed UBRank outperforms other baselines over all seven metrics. The advantage of using URL biased posts can also be seen by comparing RTRankU and RTRankA. Specifically, RTRankU only considers the posts with URLs while RTRankA considers all the posts on the graph construction. The rest of the process is the same for the two methods. The performance of RTRankU is better than RTRankA. The result is consistent with our finding that the posts with URLs have higher quality than the posts without URLs, for which the "vote" from the retweet of the posts without URLs is not that accurate for measuring user authority.

From Table 1, we can also find that the performance of RTRankA is comparable to the performance of TwitterRank, which indicates that the following relationship has almost the same effect for measuring user quality. The performance of TuRank is better than RTRankA and TwitterRank shows that by combining the following information and retweet information improvement can be achieved. The most effective information for evaluating user quality is the reweet action of the posts with URLs: RTRankU outperforms RTRankA, TwitterRank at the same time. This indicates that the retweet information based on the posts with URLs (URL biased) is the most effective beating the following

Table 1. Performances based on Top 10 users

Model	Kendall's τ
UBRank	**0.9449**
RTRankU	**0.8436**
RTRankA	0.8167
TuRank	**0.8680**
TwitterRank	0.8103
Follower_Count	0.6077
URL_Count	0.6474
URL_RT_Count	0.6436
ALL_RT_Count	0.618

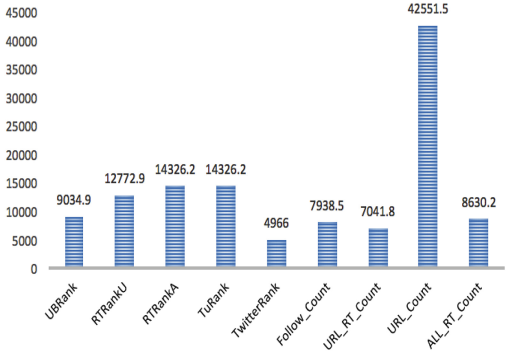


Fig. 3. Average post number

information and retweet information of all the posts on measuring user quality. Also, considering the quantity scale, the posts with URLs is a promising resource.

Table 2. List of top 10 users (with quality score)

UBRank(score)	**RTRankA(score)**	**TuRank(score)**	**TwitterRank(score)**
爱可可-爱生活(11,849)	爱可可-爱生活(11,849)	百度(455)	王斌_IIEIR(3,161)
好东西传送门(6,300)	百度(455)	爱可可-爱生活(11,849)	李航博士(1,172)
丕子(3,493)	199IT-互联网数据中心(465)	199IT-互联网数据中心(465)	唐杰THU(507)
1000sprites(3,919)	互联网分析沙龙(1,057)	互联网分析沙龙(1,057)	余凯_西二旗民工(810)
LR机器学习计算机视觉(2,308)	刘江总编(882)	刘江总编(882)	刘挺(583)
gootobe(2,880)	梁斌penny(1,389)	梁斌penny(1,389)	刘知远THU(3,058)
duin_shawe(3,018)	开源中国(4,206)	开源中国(4,206)	孙茂松(308)
数据娃掘-刘壮(1,709)	EMC中国研究院(1,886)	EMC中国研究院(1,886)	梁斌penny(1,389)
小诺_Noah(1,462)	CSDN云计算(871)	张栋_机器学习(2,024)	刘康_自动化所(1,512)
IBM_Huiwen_Watson(2,197)	张栋_机器学习(2,024)	CSDN云计算(871)	好东西传送门(6,300)

Table 2 present the top 10 users generated by different methods. The score here is the standard user quality score based on the human judgments. Specifically, UBRank can find users who may not be very famous but they do update high quality contents; RTRankU and RTRankA have similar results but different preferences for users; As we expected, TwitterRank can find authority users who are quite famous in the studied domain. The advantage of our UBRank is that it can find the users who may not be very famous offline but they accumulate many high quality contents (URLs) which are well accepted in the studied domain. An example is the user "1000sprites", who only appears in our top 10 results. He has only 119 followers, but the sum of its post quality is up to 3,919.

We also present the average number of the posts for the top 10 users (Fig. 3) generated by different methods. Obviously, our method UBRank is not aggressive to pick up the users with the most posts but with the posts at a moderate scale due to the consideration of post quality.

6 Conclusion

In this paper, we propose to use only posts with URLs to compute user quality, where the hub factor and authority factor are both important for measurement. Specifically, we propose a graph based iterative algorithm called UBRank to rank microblog users. Experiments demonstrate the advantage of using URL biased posts and the effectiveness of the proposed UBRank for measuring user quality, which achieves a high consistence with human judgments.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (grant No. 61402466 and 61572494) and the Strategic Priority Research Program of the Chinese Academy of Sciences (grant No. XDA06030200).

References

1. Pal, A., Counts, S.: Identifying topical authorities in microblogs. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 45–54. ACM (2011)
2. Yamaguchi, Y., Takahashi, T., Amagasa, T., Kitagawa, H.: TURank: Twitter user ranking based on user-tweet graph analysis. In: Triantafillou, P., Suel, T., Chen, L. (eds.) WISE 2010. LNCS, vol. 6488, pp. 240–253. Springer, Heidelberg (2010)
3. Bakshy, E., Hofman, J.M., Mason, W.A., et al.: Everyone's an influencer: quantifying influence on Twitter. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 65–74. ACM (2011)
4. Lee, C., Kwak, H., Park, H., et al.: Finding influentials based on the temporal order of information adoption in Twitter. In: Proceedings of the 19th International Conferenceon World Wide Web, pp. 1137–1138. ACM (2010)
5. Romero, D.M., Galuba, W., Asur, S., Huberman, B.A.: Influence and passivity in social media. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011, Part III. LNCS, vol. 6913, pp. 18–33. Springer, Heidelberg (2011)
6. Cha, M., Haddadi, H., Benevenuto, F., et al.: Measuring user influence in Twitter: the million follower fallacy. ICWSM **10**(10–17), 30 (2010)
7. Tunkelang D.: A Twitter analog to pagerank. The Noisy Channel (2009)
8. Weng, J., Lim, E.P., Jiang, J., et al.: TwitterRank: finding topic-sensitive influential twitterers. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 261–270. ACM (2010)
9. Kong, S., Feng, L.: A tweet-centric approach for topic-specific author ranking in micro-blog. In: Tang, J., King, I., Chen, L., Wang, J. (eds.) ADMA 2011, Part I. LNCS, vol. 7120, pp. 138–151. Springer, Heidelberg (2011)
10. Gupta, P., Goel, A., Lin, J., et al.: WTF: the who to follow service at Twitter. In: International World Wide Web Conferences Steering Committee on Proceedings of the 22nd International Conference on World Wide Web, pp. 505–514 (2013)
11. Ghosh, S., Sharma, N., Benevenuto, F., et al.: Cognos: crowdsourcing search for topic experts in microblogs. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 575–590. ACM (2012)

12. Suh, B., Hong, L., Pirolli, P., et al.: Want to be retweeted? Large scale analytics on factors impacting retweet in Twitter network. In: 2010 IEEE Second International Conference on Social Computing (SOCIALCOM), pp. 177–184. IEEE (2010)
13. Antoniades, D., Polakis, I., Kontaxis, G., et al.: we.b: The web of short URLs. In: Proceedings of the 20th International Conference on World Wide Web, pp. 715–724. ACM (2011)

Identifying Implicit Enterprise Users from the Imbalanced Social Data

Zhenni You, Tieyun Qian(✉), Baochao Zhang, and Shi Ying

State Key Laboratory of Software Engineering, Wuhan University, Wuhan, China
{znyou,qty,bczhang,yingshi}@whu.edu.cn

Abstract. Identifying the implicit enterprise users in social media enables the improvement of data quality for many applications like user profiling and targeted advertisement, as they register as ordinary users but act like enterprise ones and hence become the noises in the data. The recognition of implicit enterprise users confronts two challenges: (1) it needs to be handled quickly with little cost due to the very nature of preprocessing, and (2) it is necessary to deal with the highly skewed distribution of implicit enterprise users and ordinary users, which is about 1:10 in a social media site Sina Weibo in China. To the best of our knowledge, this problem is so far unexplored.

In this paper, we present an efficient class-imbalance learning framework which involves several types of new features from the users' profile. Specifically, a cost sensitive learning strategy is designed to overcome the problem arising from the skewed data, and a set of novel features are extracted from the profile rather than the main contents to greatly reduce the overhead of crawling and processing the microblogs. We conduct extensive experiments on a real data set consisting of 2200 users (2000 ordinary users and 200 implicit enterprise users, respectively) in Sina Weibo. The results demonstrate that our method significantly outperforms the baselines by a large margin.

Keywords: User classification · Implicit enterprise users · Feature extraction · Imbalanced data

1 Introduction

There are a special group of enterprise users in social media like Sina Weibo in China, who register as *ordinary users* (OUs) but act like enterprise ones. In other words, the account is mainly used for brand or product promotion rather than the personal interaction with other people. We call this kind of users as *implicit enterprise users* (IEUs). Although Sina Weibo defines a special user type, known as Blue V, which we call them *explicit enterprise users* (EEUs), many small enterprises do not choose to be verified as the Blue V users due to the strict or long verification procedure. They still register as ordinary users and become implicit enterprise users. The account of an implicit enterpriser user can stand for an object like a shop, a company, or a society rather than a human

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 94–101, 2016.
DOI: 10.1007/978-3-319-48743-4_8

being. It will be harmful if we include implicit enterpriser users into the corpus when doing user profiling or demographic analysis [3,5,6,11]. Thus it is necessary to eliminate the implicit enterprise users before starting such tasks. In this paper, we treat it as a two-class classification problem, one for implicit enterprise users and the other for ordinary users.

Due to the similarity to the EEUs (Blue V), the IEUs can be recognized by simply using a set of high-frequency words like “company”, “center”, and “community”. However, our preliminary investigation finds that many IEUs do not follow this rule. For example, a user with the name of ‘the alliance of dream chasing” is a ordinary user while another username “Feeling the beauty of painting” looks like a nickname of a sentimental person, but it actually belongs to an implicit enterprise user. We will show in the experiment part that the performance of such a straight-forward approach is quite poor. Moreover, by analyzing the application scenario of the task of implicit enterprise user, we find that (1) this task is usually regarded as a preprocessing component and needs to be handled quickly with little cost, and (2) the ratio of IEUs to OUs is highly skewed, i.e., about 1/10 in a sample we investigate.

Based on the above observations, we present an efficient class-imbalance learning framework which involves several types of new features extracted from the users’ profile. In particular, a cost sensitive learning strategy is designed to overcome the problem incurred by the skewed data, and the profile features rather than the content features are used to greatly save the time overhead of crawling and processing the microblogs.

The contributions of this paper are as follows:

1. We present a first-of-a-kind problem for detecting implicit enterprise users who are a special type of noises in social media.
2. We extract a set of new features from the user profile which capture the intrinsic properties of the implicit enterprise users.
3. We design an efficient class-imbalance learning framework for accurately recognizing the implicit enterprise users.

2 Related Work

Real word data are usually imbalanced, i.e. some classes have much more samples than others. The imbalanced learning problem has drawn a great amount of studies [4,12]. The most commonly used approach is the sampling method which alters the sizes of training sets. More sophisticated sampling like SMOTE [1] carefully create artificial data considering the relationship between examples. The cost-sensitive learning considers the costs associated with misclassifying examples [8,9]. It uses different weights to represent the costs for misclassifying any samples in the data. Overall, the cost-sensitive learning is more efficient and is often superior to sampling methods in terms of classification performance. Hence we adopt it to solve our problem.

Spammer detection is another related field to our study. Finding good features is a main focus in this area. Most of existing studies used the content features including duplicate reviews [2,10], the urls [15], the posting relations [13], profile features such as the ratio of sent/acceptance invitation [14]. In general, the features useful for spammer detection may be ineffective in recognizing enterprise user. In this paper, we present a number of new features for effectively recognizing implicit enterprise users.

3 Overall Framework for Identifying Implicit Enterprise Users in Imbalanced Data

3.1 Novel Profile Features

Sina Weibo allows users to register their basic information in the profile and post microblogs in his/her homepage. Developers are provided with APIs to access the profile and microblogs. Normally, the profile information is easier to get than the microblogs since it costs much more requests to access the microblogs. If we can remove most of the implicit enterprise users merely using the profile information, the resources will be greatly saved for later processing. For the fast identification of implicit enterprise users, we use features from the profile instead of using features from microblogs.

We extract six types of features from users' profiles. The privacy features reflect whether the user would disclose his/her personal information such as blood type and profession to the public. The contact features measure the degree to which the user would connect to the others. The personalized features describe users' tags and description. The status features are assigned by Sina, which record how frequently a user logs into the system. Both the friend constitution and character n-gram features are proposed to analyze how the user's screenname and interests are related to different types of users.

3.2 Class-Imbalance Learning Approach

After extracting the profile features, each user can be represented as a vector in the designated feature space. We then follow a class-imbalance learning framework for detecting the implicit enterprise users in the imbalanced social data.

The objective of class-imbalance learning is to improve the identification performance on the minor (positive) class. By associating the positive samples with a higher value, the cost sensitive learning denotes a higher importance of correctly identifying these samples. Since the cost sensitive learning only changes the weight for each sample in the data set, it can be combined with any supervised learning algorithms such as support vector machine (SVM) or logistic regression (LR). SVM needs a careful tuning to reach high performance. In contrast, LR is less sensitive to the parameters and is more efficient than SVM in most of the cases. Hence we use LR as our classifier.

We briefly describe the basic concepts in two-class LR classification. Let $L = \{(x_i, y_i)\}$ (i = 1..N) be a set of training samples, where each $x_i \in R^n$ is the feature vector of user u_i; y_i is a class label in Y = {0,1}. The logistic regression assumes the following probability model: $P(y|x,w,b) = \frac{1}{1+e^{-y(w^Tx+b)}}$, where $w \in R^n$ and $b \in R$ are the parameters of the model. For a simpler derivation, we omit the bias term b and represent $[w^T;b]$ as $[w^T]$, and then the model is defined as:

$$P(y|x,w) = \frac{1}{1+e^{-y(w^Tx)}} \tag{1}$$

To solve the problem is to find parameters fitting well for the training data. Usually the parameters are estimated using the likelihood function $\prod_{i=1}^{N} P(y_i|x_i,w)$. It can be transformed to maximize the log-likelihood function, which is defined as:

$$\max_w L(w) = -\sum_{i=1}^{N} log(1+e^{-y_i*(w^Tx_i)}) \tag{2}$$

It can be further transformed into the dual problem to minimize the negative log-likelihood:

$$\min_w f(w) = \sum_{i=1}^{N} log(1+e^{-y_i(w^Tx_i)}) \tag{3}$$

In order to enhance the generalization ability, we add a L_2 regularization factor on $f(w)$. The objective for LR on balanced data is defined as:

$$\min_w f_b(w) = (\frac{1}{2}w^Tw + \lambda\sum_{i=1}^{N} log(1+e^{-y_i(w^Tx_i)})) \tag{4}$$

where λ is a parameter to balance the two terms in Eq. 4.

Suppose that we have $L^+ = \{(x_i,y_i)\}$ (i = 1..N^+) and $L^- = \{(x_j,y_j)\}$ (j = 1..N^-) minor and major samples, respectively. For a class-imbalance problem where $N^+ \ll N^-$, we need differentiate the samples in the minor and major class. Hence we associate a weight of cost item $c_i \in$ [0, r] for each training example in L^+ and L^-. For simplicity, we assign the same costs to all the minor (major) samples. The objective for LR on imbalanced data is then defined as:

$$\min_{w,b} f_i(w) = (\frac{1}{2}w^Tw + \alpha\sum_{i=1}^{N^+} log(1+e^{-y_i(w^Tx_i)}) + \beta\sum_{j=1}^{N^-} log(1+e^{-y_j(w^Tx_j)})) \tag{5}$$

where α and β is the cost for minor and major samples, respectively.

Both Eqs. 4 and 5 are the optimization problem and can be solved by a number of approaches like gradient descent or Newton method. In this paper, we adopt the Newton method. The detailed steps for the overall framework for class-imbalance learning are given in Algorithm 1.

In Algorithm 1, line 1 initializes the variables. Lines 2 to 8 are used to update the parameters in an iterative way. Line 9 assigns the parameters using the values returned from the last round. Line 10 computes the probability of y_i for t_i. Lines 11 to 15 finish the class label assignments for test samples.

Algorithm 1. $IEUFinder^{imb}$

```
Require: the minor and major training data set $L+$ and $L-$, a test data set $T = \{t_1, t_2, ...t_n\}$, the objective function $f_i(w)$, the gradient $g(w) = \nabla f_i(w)$, the Hessian matrix $H(w)$ of $f_i(w)$, the termination criterion $\varepsilon$.
Ensure: the class label assignment for each $t_i$ in $T$.
Initialize $w^0$, $g_0 = g(w^0)$, set $k = 0$
while $||g_k|| >= \varepsilon$ do
    Compute $g_k = g(w^k)$
    Compute $H_k = H(w^k)$
    Compute $s_k = -{H_k}^{-1} g_k$
    $w^{k+1} = w^k + s_k$
    $k = k + 1$
end while
$w = w^k$
Compute the probability for $t_i$ using equation Eq. 1.
if $P(y = 0|t_i, w) > P(y = 1|t_i, w)$ then
    $y(t_i) = 0$
else
    $y(t_i) = 1$
end if
```

4 Experimental Evaluation

4.1 Experiment Setup

We conducted experiments on a real data set from Sina Weibo. It contains 200 IEUs and 2000 OUs, respectively. We conduct 5 fold cross-validation. The results are averaged over five folds. We report the F1 score for the minor class as the evaluation metric. Since no existing works are tailed for our task, we propose the following six baselines for comparison.

(1) *MatchBV*: We sort the character n-grams (n = 2, 3) of the screennames of Blue V users in descending order, and then choose a certain percent of character n-grams to match the screenname in the test set. If matched, then we label the test data as an implicit enterprise user.

(2) *MatchOU*: We sort the character n-grams (n = 2, 3) of the screennames of ordinary users in descending order, and then choose a certain percent of character n-grams to match the screenname in the test set. If matched, then we label the test data as an ordinary user.

(3) *NaiveBayes(NB)*: We first calculate the probability of each character n-gram (n = 2,3) in Blue V and ordinary users. Then we sum the probability values of all the character n-grams in one user's screenname. The user is assigned a label with a larger value.

(4) $IEUFinder^{bal}$: We use the same profile features as those used in $IEUFinder^{imb}$. The model is also the logistic regression with L2 regularization. The only difference is that $IEUFinder^{bal}$ does not distinguish the costs for samples in minor and major classes, i.e., $\alpha = \beta = 1$.

(5) *SplitData*: We apply the under-sampling strategy on the data, i.e., the samples in the major class are randomly split into n parts. Then a LR with L2 regularization classifier is applied to each partition. Finally the results from the n classifiers are further ensembled to get better performance [7].
(6) *SplitFeature*: We apply under-sampling strategy on the features. All the other settings are similar with *SplitData*.

Please note that the first four baselines are classic learning methods and the last two baselines are class-imbalance learning.

4.2 Effects of Cost Items

We then evaluate the performance of $IEUFinder^{imb}$ by varying the ratio of cost item α (for minor class) to β (for major class). The results are shown in Fig. 1. The left and right sub-figures show the effects of cost items for $\alpha > \beta$ and $\alpha < \beta$, respectively.

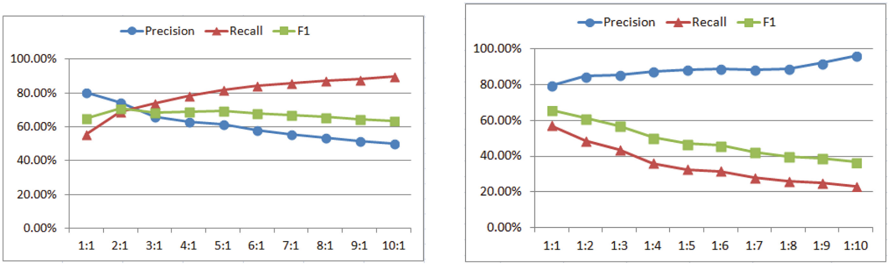


Fig. 1. Effects of cost items α and β

It is clear that setting high cost weight on major class (the right sub-figure) severely damages the performance. The F1 curve is very steep, its value decreasing from 64.88 % to 36.77 %. In contrast, the F1 curve in the left sub-figure is steady. Overall, with the increase ratio of α:β ($\alpha > \beta$), the recall continuously ascends and the precision descends. This is intuitive as more users are recognized as IEUs with larger weight on minor class. The best F1 value (70.71 %) is got when α:β is set to 2:1, showing a 5.83 % improvement over the start point. In the following, we will use α:$\beta = 2$:1 as our default setting.

4.3 Comparison with Baselines

We now compare our proposed algorithm $IEUFinder^{imb}$ with the six baselines. The results are presented in Table 1. For $MatchBV$ and $MatchOU$, we select the top 0.001, 0.005, 0.01, 0.05, 1, 5, and 10 percent of high frequency words from the EEUs and the OUs, respectively, for matching the test cases, and then select the best results for comparison. For *SplitData* and *SplitFeature*, we also vary n from 1 to 10 and present their best results.

Table 1. Comparison with baselines

Approaches	Precision	Recall	F1
MatchBV	22.57	66.50	33.68
MatchOU	10.97	96.00	19.68
NB	24.17	81.00	37.16
$IEUFinder^{bal}$	80.21	55.50	64.88
SplitData	45.46	87.50	59.24
SplitFeature	45.00	88.50	59.13
$IEUFinder^{imb}$	**74.28**	**69.00**	**70.71**

We have the following important notes.

- The F1 score for *MatchOU* is only 19.68 %, the worst among all methods. This is because many of the character n-grams in the ordinary users are widely used. They can be easily matched by both implicit enterprise users and ordinary users, resulting in all users are classified as ordinary ones.
- *NB* is better than *MatchBV* and *MatchOU*. This infers that using the distribution of character n-grams in the screenname is more reasonable than using a straight-forward matching strategy.
- $IEUFinder^{bal}$ performs the best among four traditional baselines with a 64.88 % F1 score. This suggests that the multiple types of profile features are more effective than the character n-gram features used in other three baselines.
- Two under-sampling methods *SplitData* and *SplitFeature* are worse than $IEUFinder^{bal}$, showing that an improper class-imbalance learning approach may hurt the performance. Another finding is that while their F1 scores are worse than that of $IEUFinder^{imb}$, their recall values are much higher. This is intuitive because when the data or features are split, the gap between the size of the minor and major class is reduced. Hence samples in minor class are easier to be found.
- $IEUFinder^{imb}$ is the best. It is much better than all four classic learning approaches, and it also significantly outperforms two under-sampling methods, showing a huge increase on F1. This clearly demonstrates that our cost sensitive learning approach is very effective in dealing with the introduced imbalanced learning problem.

5 Conclusion

We introduce a new research problem of identifying implicit enterprise users in social media. This problem is expected to be solved efficiently in a skewed data distribution. To this end, we present a class imbalance learning framework which involves several types of new features extracted from users' profile. The cost-sensitive classification is effective in handling imbalanced data. By using the profile features rather than the content ones, we greatly save the costs in crawling and processing the microblogs. We conduct extensive experiments on a real data set. Results demonstrate that our proposed method is very effective in recognizing implicit enterprise users. It also significantly outperforms all the baselines with a large margin.

We wish this study will initiate the research of the noise removal in the area of social media processing. In the future, we plan to explore more features to further improve the performance and compare our problem with the task of spammer detection to see if they can be put into a unified framework.

Acknowledgements. The work described in this paper has been supported in part by the NSFC projects (61572376, 61272275, 61373038), and the 111 project (B07037).

References

1. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: Smote: synthetic minority oversampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
2. Chen, C., Wu, K., Srinivasan, V., Zhang, X.: Battling the internet water army: detection of hidden paid posters. In: Proceedings of ASONAM, pp. 116–120 (2013)
3. Filippova, K.: User demographics and language in an implicit social network. In: Proceedings of EMNLP, pp. 1478–1488 (2012)
4. He, H., Garcia, E.A.: Learning from imbalanced data. TKDE **21**(9), 1263–1284 (2009)
5. Kosinski, M., Stillwell, D., Graepel, T.: Private traits and attributes are predictable from digital records of human behavior. PNAS **110**, 5802–5805 (2013)
6. Li, J., Ritter, A., Hovy, E.: Weakly supervised user profile extraction from Twitter. In: Proceedings of ACL, pp. 165–174 (2014)
7. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory under-sampling for class-imbalance learning. In: Proceedings of ICDM, pp. 965–969 (2006)
8. Liu, X.Y., Zhou, Z.H.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. TKDE **18**, 63–77 (2006)
9. McCarthy, K., Zabar, B., Weiss, G.: Does cost-sensitive learning beat sampling for classifying rare classes? In: Proceedings of International Workshop Utility-Based Data Mining, pp. 69–77 (2005)
10. Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., Castellanos, M., Ghosh, R.: Spotting opinion spammers using behavioral footprints. In: Proceedings of KDD, pp. 632–640 (2013)
11. Nguyen, D., Trieschnigg, D., Doğruöz,, A.S., Grave, R., Theune, M., Meder, T., de Jong, F.: Why gender and age prediction from tweets is hard: lessons from a crowdsourcing experiment. In: Proceedings of COLING, pp. 1950–1961 (2014)
12. Sun, Y., Kamel, M.S., Wong, A.K., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. Pattern Recogn. **40**, 3358–3378 (2007)
13. Wu, F., Shu, J., Huang, Y., Yuan, Z.: Social spammer and spam message codetection in microblogging with social context regularization. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 1601–1610 (2015)
14. Yang, Z., Wilson, C., Wang, X., Gao, T., Zhao, B.Y., Dai, Y.: Uncovering social network sybils in the wild. ACM Trans. Knowl. Discov. Data **8**, 2 (2014)
15. Zhang, X., Li, Z., Zhu, S., Liang, W.: Detecting spam and promoting campaigns in Twitter. ACM Trans. Web **10**(1), 4:1–4:28 (2016)

Microblog Data Analysis

Understanding Factors That Affect Web Traffic via Twitter

Chunjing Xiao[1,2,3](✉), Zhiguang Qin[2], Xucheng Luo[2], and Aleksandar Kuzmanovic[3]

[1] School of Computer and Information Engineering,
Henan University, Kaifeng, China
chunjingxiao@gmail.com

[2] School of Information and Software Engineering, UESTC, Chengdu, China
{qinzg,xucheng}@uestc.edu.cn

[3] Department of EECS, Northwestern University, Evanston, USA
akuzma@cs.northwestern.edu

Abstract. Currently, millions of companies, organizations and individuals take advantage of the social media function of Twitter to promote themselves. One of the most important goals is to attract web traffic. In this paper, we study the problem of obtaining web traffic via Twitter. We approach this problem in two stages. First, we analyze the correlation between important factors and the click number of URLs in tweets. Through measurements, we find that the commonly accepted method, increasing followers by reciprocal exchanges of links, has limited effects on improving the number of clicks. And characteristics of tweets (such as the presence of hashtags and tweet length) exert different impacts on users with different influence levels for obtaining the click number. In our second stage, based on the analyses, we introduce the Multi-Task Learning (MTL) to build a model for predicting the number of clicks. This model takes into account the specific characters of users with different influence levels to improve the predictive accuracy. The experiments, based on Twitter data, show the predictive performance is significantly higher than the baseline.

Keywords: Popularity · Prediction · Web traffic · Twitter

1 Introduction

Web traffic is one of the key indicators of a website's success, and most of individuals and companies rank websites mainly on the basis of their web traffic, such as the well-known Alexa[1]. Thus, website owners constantly strive to increase their web traffic by implementing various strategies, such as advertisements or audience analyses. The popularity of Twitter provides a new mean of promoting websites. In fact, Twitter has become new influential media for information sharing [18]. Thus, millions of organizations, companies, and individuals register

[1] http://www.alexa.com/.

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 105–120, 2016.
DOI: 10.1007/978-3-319-48743-4_9

accounts on that and publish their URLs to attract web traffic, and Twitter has been a beneficial platform for a number of the websites [3,21].

Although the capability of Twitter to generate web traffic is widely accepted, little work focuses on examining the factors that affect obtaining web traffic via Twitter, and a serial of questions in this field keep unknown. For example, the previous work shows that the number of followers does not necessarily reflect their influence in terms of retweets or mentions [8], however, the reason is still unknown. To increase the follower number, users may randomly follow others in the hope that they follow back [23], and this phenomenon is called reciprocal links by Ghosh *et al.* [12]. Whereas it is not clear whether these types of followers can enhance content diffusion. In addition, there is a need to understand how hashtags and mentions in tweets impact the click number of URLs and whether these factors have the predictive power of the click number.

Our approach to answering these questions begins with an extensive characterization of important affecting factors, such as the follower number, presences of hashtags and mentions, as well as tweet length. To understand the impact of followers, we analyze the correlation between the numbers of clicks and followers, and find their correlation is not as strong as expected, which is consistent with the finding in [8]. However, the difference in the numbers of followers and reciprocal links has an obviously higher coefficient of correlation with the number of clicks. Therefore, reciprocal links are a key reason why the number of user followers does not necessarily reflect their influence in terms of the click number. And our further analyses also show reciprocal links have limited effects on content diffusion, although it is widely used to increase the number of followers.

Besides, we exploit the effect of tweet characteristics on the click number, such as the presences of hashtags and mentions and tweet length. And we find that the correlation between the number of clicks and these characteristics exhibits different trends for users with different influence levels (Here the influence level is measured by the difference between the numbers of followers and reciprocal links). Specifically, in terms of hashtags, URLs in tweets with hashtags obtain more clicks for users with low influence, but less for users with high influence. And for tweet length, when tweets have 50 and 120 characters, their URLs attract a similar maximum number of clicks for users with low influence. However, it is hardly affected by tweet length for users with high influence.

The second part of work for answering these questions is to conduct prediction about the number of clicks. Because the above analyses show that hashtags, mentions and tweet length exert different effects on users with different influence levels for obtaining the number of clicks, the model should take into account these different effects to improve predictive performance. To this end, we cast the predictive problem as a Multi-Task Learning (MTL) problem.

Specifically, we build a SVM+MTL model to predict the number of clicks. In this model, users are placed into different groups based on their influence levels, and each group is treated as a task. The model considers both the common properties of all the users and specific characters of users with different influence levels to improve predictive performance. Based on the Twitter data, the experiment results show the accuracy of our model is significantly higher than the baseline.

2 Related Work

There is little work focusing on the number of clicks on Twitter, however, the number of clicks, to some extent, can be a measure of popularity. Therefore, our work is related to the fields of popularity, which mainly consist of two threads of work: analyzing factors that affect popularity and predicting popularity in social media.

For the analyses of affecting factors, Suh *et al.* [24] examine a number of features that might affect the retweets. They find that URLs, hashtags and the numbers of followers and friends affect the retweets. Comarela *et al.* [10] identify factors that influence user response or retweet probability. They find that some basic textual characteristics, such as message size and the presence of hashtags, mentions and URLs, affect the replies or retweets. Liu *et al.* [20] evaluates eleven extrinsic factors that may influence the response rate in social question and answering from Sina Weibo. They show that the features, such as the number of followers, frequency of posting, hashtags and emotion, can be used to predict the number of responses. Apart from microblogs, Khosla *et al.* [16] and Bakhshi *et al.* [4] study the important factors that impact the popularity of images and quality of reviews respectively. Compared with these studies, we, beyond analyzing basic factors, explore the reason of existed phenomenons, and study whether tweet characteristics (such as hashtags, mentions and tweet length) exert different impacts on URLs in tweets of users with different influence.

For the popularity prediction, the studies fall into two main genres: conducting prediction before and after content publication. For the former, because the distribution of cascade sizes is very skewed, predicting the exact number of cascade sizes remain relatively unreliable [5]. Hence, rather than predicting exact integer values, most of the researchers define several categories to represent the popularity levels and predict which categories contents will belong to. For example, Hong *et al.* [13] define several categories to represent popularity of tweets and use logistic regression to predict the categories of tweets. Jenders *et al.* [15] predict whether a given tweet will be more frequently retweeted than a certain threshold. They firstly analyze the correlation between the retweet frequency and user features, and then they use the probabilistic models to conduct prediction. Vasconcelos *et al.* [27] categorize reviews into various popularity levels and predict the levels using multivariate linear regression and SVM models.

To achieve higher accuracy of prediction, many studies predict popularity after content publication. In this case, the early number of retweets or views within a short period after content publication can be used for prediction. Some work uses the early information to predict the exact integer values. For example, Szabo *et al.* [25] find the early number of retweets or views is strongly correlated with the later number on Digg and YouTube, and predict the popularity of content based on this finding. Kupavskii *et al.* [17] and Bao *et al.* [7] improve the performance of popularity prediction by exploiting the features of the cascade flow and structural characteristics respectively And Zhao *et al.* [28] develop a self-exciting Point Process Model to predict tweet popularity.

Other work still uses the early information to predict the categories which represent the popularity levels. For example, Gao *et al.* [11] predict whether a tweet will be popular based on temporal features of first 10 retweets using the bagged decision trees model. Given a cascade that currently has size k, Cheng *et al.* [9] predict whether it grow beyond the median size 2k by using the temporal and structural features. They use a variety of learning methods, including logistic regression classifier, naive Bayes and SVM for the prediction.

The method of popularity prediction after content publication generally achieves better performance than that of before content publication, but it is still crucial for the prediction before content publication. Because (*i*) publishers always want to know popularity of their contents before publication, (*ii*) and this method can clearly measure the importance of static factors in affecting popularity. Therefore, we conduct prediction before content publication. And our MTL-based predictive model is built based on our findings. To the best of our knowledge, we are the first to predict popularity using MTL.

3 Data Description

3.1 Background of URL Clicks

In this section, we present information about clicks of short URLs. Due to the limitation of tweet length, users tend to publish shortened URLs on Twitter. Therefore, the service of shortening long URLs is provided by many companies, and Bitly is among the most popular ones. Furthermore, Bitly APIs[1] provide the information about the click number of URLs in tweets. These number can be classified into two types: the *exact click number* referring to the number of clicks from a given tweet of the user; the *global click number* referring to the number of clicks from all the domains and platforms, including Twitter, Facebook and so on. For these two kinds of numbers, the exact click number can be considered as the ability of the tweet to attract web traffic. Therefore, the exact click number are used as the standard for analyzing factors that affect web traffic via Twitter. And the global click number can be used to reflect the popularity of the tweet content, and will be used as one of the features to predict the exact click number. Below the click number will refer to the exact click number for simplicity.

3.2 Twitter Data

As our goal is to analyze users who are aiming to attract web traffic via Twitter, we need to select users who tend to publish tweets with short URLs. In our study, we only select short URLs hosted by Bitly, because their exact click number can be obtained, and they are the most popular ones, taking about 50 % of all the URLs in Twitter [3].

[1] http://dev.bitly.com/api.html.

To select targeted users, we firstly extract domains hosted by Bitly based on a random sample of public tweets (around 790 million) collected by Twitter streaming APIs. And we obtain 6,524 domains hosted by Bitly, including many well-known companies and organizations, such as nyti.ms (New York Times), wapo.st (Washington Post) and es.pn (ESPN). Secondly, from these 790 m tweets, we extract the users whose language is English and whose tweets include at least one short URL hosted by Bitly. Base on this, we further select users who tend to publish Bitly URLs and tend to increase their websites via Twitter. According to these rules, we select users whose ratios of Bitly URLs are more than 50 %, and whose domain focuses are more than 50 %. Here the domain focus is defined as the degree of short URLs redirecting to the same domain, and can be calculated as follows: $D_i = \frac{1}{V_i} \max_k v_{ik}$, where V_i refers to the summary of URLs of user i, and v_{ik} refers to the number of URLs with the domain k of user i. If all the URLs published by a user redirect to one domain, its domain focus will be 1. Finally, 214,293 users are selected as our targeted users.

For these selected users, by Twitter APIs, we download their profiles, followers, and friends, as well as their tweets during June 2014, as shown in Table 1. And by Bitly APIs, we collect the click information of short URLs extracted from these tweets.

Table 1. Summary of Twitter data

Number of users	214,293
Number of follower links	1,261,721,039
Number of friend links	180,803,547
Number of tweets	46,286,824
Number of short URLs	34,338,613

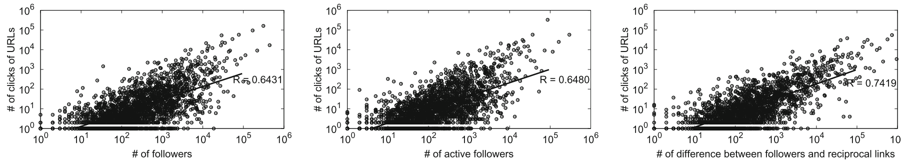
4 Analyses of Affecting Factors

We firstly describe the effect of user followers and tweet characteristics on the click number. The results in the section are the foundation for the predictive method, which is presented later.

4.1 The Role of User Followers

The number of followers is frequently used to gauge influence or reputation of users [14, 23], and compare to other criterions, such as the number of retweets and mentions [8, 18]. Therefore, we first analyze how the number of followers is correlated with the number of clicks received by URLs in tweets.

Figure 1(a) shows the correlation between the numbers of followers and URL clicks. The X-axis is the number of user followers, and the Y-axis is the sum of



(a) Followers (b) Active followers (c) Difference in followers and reciprocal links

Fig. 1. The correlation between the numbers of followers and clicks

the number of clicks. This figure shows that the coefficient of linear correlation, 0.64, is not as high as expected. This finding is consistent with the previous work, which shows that popular users with a high number of followers do not necessarily have high influence in terms of retweets or mentions [8] and the global click number of short URLs [22].

This observation raises the question why the number of followers is not very strongly correlated with the number of clicks. To address this question, we conduct analyses from two perspectives. (*i*) How do inactive followers affect the relationship between the numbers of followers and clicks? Thomas *et al.* [26] show that numerous accounts on Twitter have been suspended because of spamming issues or similar reasons. Moreover, some users tend to register multiple accounts but use only a part of them or stop using Twitter. We, therefore, attempt to evaluate the correlation between the numbers of active followers and clicks to analyze the effect of inactive followers. (*ii*) How do reciprocal links affect the relationship of the number of followers and clicks? On Twitter, a part of users randomly follow other users in the hope that they will follow back, whereas, some users join groups in which each member agrees to follow all of the other members in that group [23]. This phenomenon, which is called the reciprocal links by Ghosh *et al.* [12], is a way to increase one's number of followers, and users are recommended to increase their followers through this way to gain more web traffic [1]. However, whether these reciprocal links increase the diffusion effect of content remains unclear. Therefore, we attempt to explore the correlation between the numbers of reciprocal links and clicks to answer these questions.

To analyze the effect of inactive followers, we first identify whether a user is active. In general, Twitter regard users who log in at least once a month as active ones [2]. However, considering that we cannot obtain information about logging in activities, we regard users who publish at least one tweet, including any kind of tweets such as retweets and replies, within the last two months as active ones. After collecting the recent tweets by Twitter APIs, we can compute the active followers for each user. Further, the correlation between the numbers of active followers and clicks is plotted in Fig. 1(b). The coefficient of correlation, 0.6480, is nearly the same as that of the numbers of followers and clicks. We also analyze the correlation between numbers of active followers and all followers, and find that a strong linear relationship exists between them. These results suggest that inactive followers are not the main reason behind the moderate relationship between the number of followers and clicks.

For reciprocal links, we first collect the follower and friend list of each user, and then compute the intersection between the follower set and the friend set. This intersection is regarded as the reciprocal links. Based on this data, the correlation between the number of clicks and the difference in followers and reciprocal links is calculated, as shown in Fig. 1(c). Compared with Fig. 1(a) and 1(b), the points in Fig. 1(c) are centered around the straight line and a stronger correlation exists between the number of clicks and the difference in followers and reciprocal links. The coefficient, 0.7419, is approximately 10 % higher than that of followers and clicks. These results indicate that reciprocal links considerably affect the correlation between numbers of followers and clicks. And when reciprocal links are removed, the number of followers becomes more strongly correlated with the number of clicks.

To further evaluate the effect of reciprocal links in improving the number of clicks, we analyze the correlation between reciprocal links and clicks, as well as the correlation between reciprocal links and friends. the coefficient of the former, 0.1632, indicates that reciprocal links are not significantly correlated with clicks. The coefficient of the later is 0.9125, suggesting that most of the friends originate from reciprocal links.

Therefore, based on these analyses, we conclude that although reciprocal links are widespread to be used to increase the number of followers, they have limited effects on improving the number of clicks. And the difference of followers and reciprocal links can be a better measure of user influence. Hence, below this difference is regarded as the measure of user influence (levels), and user followers refer to this difference except Sect. 5.2.

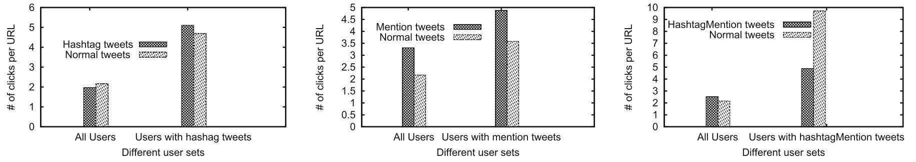
4.2 The Role of Tweet Characteristics

In this section, we analyze the impact of two kinds of tweet characteristics on the click number of URLs: tweet types (i.e., the presences of hashtags and mentions in tweets) and tweet length.

Tweet Types. On Twitter, tweets contain two widely used objects: hashtags and mentions. The former is used to mark keywords or topics in a tweet and to categorize messages, whereas the latter is a form of conversation on Twitter. Users are often encouraged to include hashtags to increase the click number of URLs on some web pages, such as [1]. Therefore, we explore how tweets that contain hashtags or mentions affect the number of clicks.

For this purpose, we group the tweets into four types: *hashtag tweets*, which are tweets that include at least one hashtag; *mention tweets*, which are tweets that include at least one mention; *hashtagMention tweets*, which are tweets that include both hashtags and mentions; and *normal tweets*, which are tweets without hashtags and mentions. To avoid any preference for users who tend (not) to publish more hashtag or mention tweets, we also analyze users with at least one hashtag, mention, or hashtagMention tweet.

Figure 2 shows the number of clicks per URL in different tweet types for different user sets. The Y-axis presents the average number of clicks for a given



(a) Hashtag tweets (b) Mention tweets (c) HashtagMention tweets

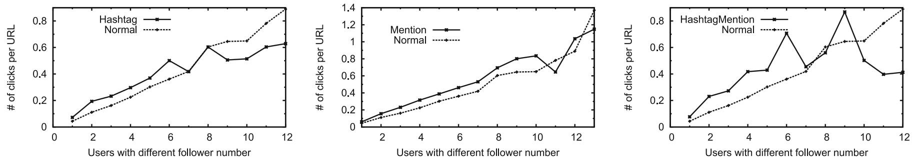
Fig. 2. Tweet type vs. number of clicks

user set. For hashtag tweets, shown in Fig. 2(a), the average number of clicks of hashtag tweets is lower than that of normal tweets for all users; however, the values are reversed for users with at least one hashtag tweet. Therefore, we cannot fully ascertain how tweets that contain hashtags correlate the number of clicks. For mention tweets, depicted in Fig. 2(b), the average number of clicks of mention tweets is always higher than that of normal tweets for both user sets. This result suggests that a positive correlation exists between tweets containing mentions and the number of clicks. For hashtagMention tweets, presented in Fig. 2(c), the trends are also inconsistent for different users.

Considering the unclear results about the effect of hashtags, we further explore whether hashtags and mentions exert the different effect on the number of clicks for users with different influence levels. For this purpose, we place users into buckets according to an interval of 200 followers. We use numbers to denote the buckets, i.e., the bucket 1 represents users with 0–200 followers, bucket 2 represents users with 200–400 followers, and so on. For each bucket, we group the tweets into four types: *hashtag tweets*, *mention tweets*, *hashtagMention tweets*, and *normal tweets*, and compute the average number of clicks for each type.

We compare the click number of the first three types of tweets with that of normal tweets, and the results are shown in Fig. 3. The figures do not show all of the buckets because of space constraints. The X-axis shows the bucket number. The Y-axis denotes the average number of clicks per URL for the particular bucket.

The results of the hashtag tweets are shown in Fig. 3(a). For the bucket 7 and 8 (referring to users with 1200–1400 and 1400–1600 followers respectively), the click numbers of the hashtag and normal tweets are very close. While, for the bucket 1 to 6, the hashtag tweets obtain a higher number of clicks than the normal tweets. However, the reverse is true for bucket 9 and beyond. These



(a) Hashtag tweets (b) Mention tweets (c) HashtagMention tweets

Fig. 3. Tweet type vs. number of clicks for users with different influence levels

results indicate that tweets with hashtags do not always achieve additional clicks, i.e., they can obtain more clicks for users with lower influence but not for that with higher influence.

For the mention tweets, presented in Fig. 3(b), from the bucket 1 to 10, the mention tweets generate a higher number of clicks than the normal tweets. While, for other buckets, the click numbers of both are interlaced with each other. That is, when users have less than roughly 1,800 followers, their tweets with mentions can attract additional clicks; however, when users have a higher number of followers, mentions do not contribute to improving the number of clicks. Affected by both hashtags and mentions, the hashtagMention tweets, presented in Fig. 3(c), exhibit a similar trend to hashtag tweets. The average number of clicks shows a small fluctuation because of their small number.

These results indicate that contrary to what people commonly assume, tweets with hashtags cannot always obtain more clicks. In fact, the hashtags and Mentions exhibit a different effect on users with different influence levels for obtaining the number of clicks.

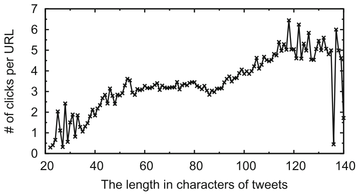


Fig. 4. All the users

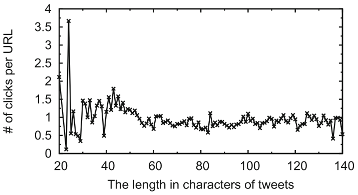
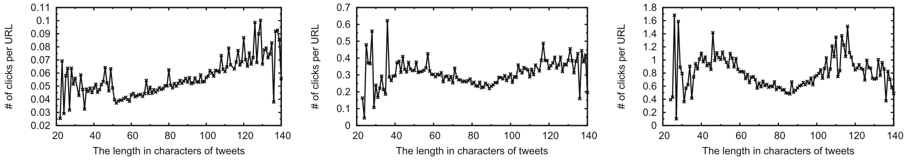


Fig. 5. Users with 2000–2200 followers

Tweet Length. Here, we explore the correlation between the tweet length and number of clicks. We first analyze this correlation for all users, and the results are shown in Fig. 4. The X-axis denotes the length of the tweets. The minimum length is 20 because the tweet contain the short URL with no less than 20 characters. The Y-axis refers to the average number of clicks with a particular length. From the figure, we can see that the number of clicks generally increases with the tweet length. And short URLs in tweets with approximately 120 characters tend to attract more clicks.

We further explore how the effect of tweet length differs for users with different influence levels. As in the previous section, we place the users into buckets according to an interval of 200 followers. For each bucket, we plot the correlations between the tweet length and number of clicks. By observing the trend of each figure, we find that these figures can be divided into two categories: users with 0–2,000 followers and users with more than 2,000 followers. For the former, all of the buckets exhibit a similar trend. In view of space constraints, we present the figures of three buckets: users with 1–200 followers, users with 600–800 followers and users with 1600–1,800 followers, as shown in Fig. 6. This category has the similar trend that the number of clicks exhibits a double hump phenomenon,



(a) users with 0-200 followers (b) Users with 600-800 followers (c) Users with 1600-1800 followers

Fig. 6. Tweet length vs. number of clicks for users with 0 to 2000 followers

and this trend becomes even more significant with the rise in the number of followers. For example, when the number of followers reaches 1,600–1,800, this trend becomes the most significant, and the two peaks of the click number are twice the minimum number of clicks. For users with more than 2,000 followers, we present the figures of users with 2,000–2,200 followers in Fig. 5, and all of the other buckets exhibit a similar trend. The number of clicks fluctuates because of the small amount of tweets when the tweet length is near 40, but it remains stable when the tweet length exceeds 50.

Basing on these results, we can conclude that the effect of tweet length on the number of clicks differs for users with different influence levels. Specifically, users with low influence, such as those with 0–2,000 followers, can be affected by tweet length, and URLs in tweets with around 50 to 120 characters tend to obtain more clicks. However, users with high influence, such as those with more than 2,000 followers, can hardly be affected by tweet length.

5 Methodology

5.1 Method of Prediction

The above analyses indicate that hashtags, mentions and tweet length place the different impact on users with different influence levels for obtaining the number of clicks. Therefore, the predictive model should take into account this different impact to achieve higher accuracy. However, a global model, such as logistic regression and SVM, will ignore this different impact. One way to address this challenge is to create and apply numerous models to the user sets with different influence levels. However, the data of some user sets, especially for user set with high influence levels, is very sparse and cannot build model accurately. Hence, to overcome this problem, we introduce the Multi-Task Learning (MTL) to predict the click number of URLs. MTL seeks to simultaneously learn the commonality as well as the differences between the multiple tasks. Therefore, we divide users into different groups based on their influence levels and treat prediction of each group as a task. And the MTL model is used to improve the performance by considering both the common properties of all users and specific characters of users with different influence levels. Here, we introduce an extension of SVM+ approach to multi task learning called SVM+MTL [19] to build the model.

In SVM+MTL, the training set T is the union of task specific sets $T_r = \{x_{ir}, y_{ir}\}_{i=1}^{l_r}$. For each task the learned weights vector is decomposed as $w + w_r, r \in (1, 2, ..., t)$ where w and w_r respectively model the commonality between tasks and task specific components. The optimization problem of SVM+MTL is formulated as follows:

$$\min_{w,b} \frac{1}{2}(w, w) + \frac{\beta}{2} \sum_{r=1}^{t} (w_r, w_r) + C \sum_{r=1}^{t} \sum_{i=1}^{l_r} \xi_{ir} \tag{1}$$

$$st : y_{ir}((w, \phi(x_{ir})) + b + (w_r, \phi_r(x_{ir})) + d_r) \geq 1 - \xi_{ir} \tag{2}$$

$$\xi_{ir} \geq 0, i = 1, ..., l_r, r = 1, ..., t \tag{3}$$

Here, all w_r's and the common w are learned simultaneously. β regularizes the relative weights of w and w_r's. ξ_{ir}'s are slack variables measuring the errors w_r's make on the t data groups. y_{ir}'s denote training labels while C regulates the complexity and proportion of nonseparable samples.

The goal of SVM+MTL is to find t decision functions $f_r(x) = (w, \phi(x)) + b + (w_r, \phi_r(x)) + d_r, r = 1, ..., t$. Each decision function f_r comprises two parts: the common weights vector w with bias term b, and the group-specific correction function w_r with bias term d_r.

5.2 Feature Spaces

In this section, we introduce the features which are used in the predictive model, including the attributes of user influence, publishing behavior, and short URLs.

Features of user influence describe the characteristics of the social topology of users. Based on the user profiles we can download by Twitter APIs, we use the metadata relative to user influence as the features, such as the number of followers, friends, lists and son on. Further, based on our analyses, we exploit the features related to influence: the active followers and differences between followers and reciprocal links, which can more accurately reflect user influence. The features are detailed in Table 2.

Features of publishing behavior are composed of the items which users can control when publishing tweets. The tweet characteristics, such as the presences of hashtags and mentions as well as tweet length, are also placed into this set, because users can determine whether their tweets include hashtags or mentions and how long their tweets are.

Features of short URLs describe the information collected by Bitly APIs. Among these features, the global click number of URLs can reflect the popularity of the tweet content, because the global click number is the sum of clicks from all the domains and platforms, and URLs in tweets are generally the key points of the tweets. The referrer number can also be a measure of popularity for URLs, because it means the sum of resources where clicks originated, i.e., the higher referrer number is, the more popular the URL is. Therefore, in the experiments later, we can evaluate whether the popularity of content has the predictive power of the exact click number by using the features about the global click number and referrer number.

Table 2. Summary of features

Feature sets	Name	Description
User influence	Followers	The number of followers
	Friends	The number of friends
	Lists	The number of lists including this user
	Active-followers	The number of active followers
	Diff-followers	Difference between followers and reciprocal links
Publishing behavior	Hashtags	The presence of hashtags in tweets
	Mentions	The presence of mentions in tweets
	Tweet length	The length of tweets
	Published time	The published time of tweets
	Average tweets	Average number of Tweets per day in our dataset
	Ratio of URLs	Ratio of numbers of tweets with URLs and all tweets
Short URLs	Global number	The global click number from all the domains and platforms
	Created time	Difference of tweet published time and URL created time
	Referrer number	The number of resources where clicks originated
	Domain ranking	Ranking in Alexa.com of the domain of expanded URLs

6 Prediction Results

Based on the method and features, we predict the click number of URLs in tweets. We describe the experiment setup and compare the results of SVM+MTL with the original SVM.

6.1 Experiment Setup

We conduct prediction before tweets publication, because compared with prediction after tweets publication, this kind of prediction can more clearly measure the factors that affect the number of clicks. As in [6,13,27], we define several categories to represent the levels of the click number and predict which categories the URL will belong to, instead of predicting the exact number. Because the latter is harder, particularly given the skewed distribution of popularity [5], and the former should be good enough for most purposes. Specifically, we divide URLs into five categories depending on the click number. That is we put URLs with 0, 1∼10, 11∼100, 101∼1,000, and more than 1,000 clicks into the category 1, 2, 3, 4 and 5 respectively. We select the same number of URLs for each category randomly, because the URLs in category 1 are dominant, accounting for around 70 % of all the URLs. When considering all the URLs for the experiments, the accuracy of prediction will reach 70 % even if we label all URLs as category 1.

The SVM+MTL takes into account both the common properties and specific characters of users with different influence levels. Hence, we place users into buckets according to an interval of 200 followers, and treat prediction of each bucket as a task. And the SVM is used as the baseline.

We use the classification accuracy and F-score to measure the performance. And the accuracy is defined as the proportion of true results in the population, and the F-score combines recall and precision with an equal weight. And to evaluate the predictive performance, we randomly divide the URLs of each user into two sets: 50 % for training and 50 % for testing.

6.2 Results and Discussion

The accuracy and F-score of the SVM and SVM+MTL predictors are presented in Table 3 for the combination of the different feature sets. The best results (biggest accuracy) for each model are emphasized in boldfaced numbers. The first observation is that although the SVM model can perform reasonably well with around 69 % accuracy using all features, the performance of SVM+MTL, 81.77 %, is significantly higher than that of SVM. Besides, no matter which feature sets are used for prediction, the accuracies of SVM+MTL are always approximately 10 % higher than that of SVM. This indicates that grouping users based on user influence levels is appropriate for SVM+MTL, and by considering both the common properties of all users and specific characters of users with different influence levels, SVM+MTL can achieve expected predictive results.

In addition, we proceed to the feature set level to determine the importance of features in predicting the click number. Unsurprisingly, for the SVM+MTL model, the accuracies of using the influence feature set and behavior feature set arrive at 74.35 % and 72.46 % respectively, which suggest that both feature sets play an important role in predicting the levels. Interestingly, the features of short URLs cannot perform as better as that of user influence and behavior. Among the short URL features, both the global click number and referrer number can, to some extent, reflect the popularity of the URL content. But they fail to have a predictive power of the exact click number. This indicates that not every user can achieve more clicks by publishing popular URLs. We also compute the coefficient of correlation between the global click number and exact click number. The lower coefficient, about 0.38, also provides support for this point.

Table 3. The predictive results

Feature sets	SVM		SVM MTL	
	Accuracy (%)	F-score (%)	Accuracy (%)	F-score (%)
Influence	65.11	64.32	74.35	73.41
Behavior	62.33	61.48	72.46	72.14
URLs	57.68	58.13	68.74	69.85
Influence + behavior	66.04	66.84	75.26	74.11
Influence + URLs	65.3	65.91	76.18	77.14
Behavior + URLs	60.27	61.05	71.49	70.28
All features	**69.49**	69.74	**81.77**	81.37

7 Conclusions

In this paper, we conducted analyses and predictions about the click number of URLs in tweets. Through the analyses, we showed that the correlation of the click numbers and followers is not as strong as expected. This is due to reciprocal links, not inactive followers. And our further analysis suggested reciprocal links have limited effects on content diffusion, although it is widely used to increase the number of followers. We also found that hashtags and tweet length place different impacts on users with different influence levels for obtaining the number of clicks. Specifically, in terms of hashtags, URLs in tweets with hashtags achieve more clicks for users with low influence, but less for users with high influence. And for tweet length, URLs in tweets with 50 and 120 characters attract a similar maximum number of clicks. However, users with higher influence are hardly affected by tweet length. Based on these analyses, we built a SVM+MTL model to predict the click number. In this model, users with different influence levels are treated as different predictive tasks, and the commonality of all users and differences of users with different influence levels are learned simultaneously. The experiments, based on Twitter data, showed our predictive performance is significantly higher than the baselines.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (No. 61402151 and 61272527), Science and Technology Foundation of Henan Province of China (No. 162102410010), and Open Research Fund of Network and Data Security Key Laboratory of Sichuan Province of China (No. NDS2015-02).

References

1. How to use twitter to increase web traffic. http://www.wikihow.com/Use-Twitter-to-Increase-Web-Traffic
2. Twitter announces 100 million active users. http://mashable.com/2011/09/08/twitter-has-100-million-active-users
3. Antoniades, D., Polakis, I., Kontaxis, G., Athanasopoulos, E., Ioannidis, S., Markatos, E.P., Karagiannis, T.: we.b: the web of short URLs. In: Proceedings of the 20th international conference on World Wide Web, pp. 715–724 (2011)
4. Bakhshi, S., Kanuparthy, P., Shamma, D.A.: Understanding online reviews: funny, cool or useful? In: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work, pp. 1270–1276 (2015)
5. Bakshy, E., Hofman, J.M., Mason, W.A., Watts, D.J.: Everyone's an influencer: quantifying influence on Twitter. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 65–74 (2011)
6. Bandari, R., Asur, S., Huberman, B.A.: The pulse of news in social media: forecasting popularity. In: The Sixth International AAAI Conference on Weblogs and Social Media, pp. 26–33 (2012)
7. Bao, P., Shen, H.W., Huang, J., Cheng, X.Q.: Popularity prediction in microblogging network: a case study on Sina Weibo. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 177–178 (2013)

8. Cha, M., Haddadi, H., Benevenuto, F., Gummadi, K.P.: Measuring user influence in twitter: the million follower fallacy. In: Proceedings of International AAAI Conference on Weblogs and Social Media (2010)
9. Cheng, J., Adamic, L., Dow, P.A., Kleinberg, J.M., Leskovec, J.: Can cascades be predicted? In: Proceedings of the 23rd International Conference on World wide web, pp. 925–936 (2014)
10. Comarela, G., Crovella, M., Almeida, V., Benevenuto, F.: Understanding factors that affect response rates in Twitter. In: Proceedings of the 23rd ACM Conference on Hypertext and Social Media, pp. 123–132 (2012)
11. Gao, S., Ma, J., Chen, Z.: Effective and effortless features for popularity prediction in microblogging network. In: Proceedings of the 23rd International Conference on World Wide Web, pp. 269–270 (2014)
12. Ghosh, S., Viswanath, B., Kooti, F., Sharma, N.K., Korlam, G., Benevenuto, F., Ganguly, N., Gummadi, K.P.: Understanding and combating link farming in the Twitter social network. In: Proceedings of the 21st International Conference on World Wide Web, pp. 61–70 (2012)
13. Hong, L., Dan, O., Davison, B.D.: Predicting popular messages in Twitter. In: Proceedings of the 20th International Conference Companion on World wide Web, pp. 57–58 (2011)
14. Hutto, C., Yardi, S., Gilbert, E.: A longitudinal study of follow predictors on Twitter. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 821–830 (2013)
15. Jenders, M., Kasneci, G., Naumann, F.: Analyzing and predicting viral tweets. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 657–664 (2013)
16. Khosla, A., Das Sarma, A., Hamid, R.: What makes an image popular? In: Proceedings of the 23rd International Conference on World Wide Web, pp. 867–876 (2014)
17. Kupavskii, A., Ostroumova, L., Umnov, A., Usachev, S., Serdyukov, P., Gusev, G., Kustarev, A.: Prediction of retweet cascade size over time. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 2335–2338 (2012)
18. Kwak, H., Lee, C., Park, H., Moon, S.: What is Twitter, a social network or a news media? In: Proceedings of the 19th International Conference on World Wide Web, pp. 591–600 (2010)
19. Lichen, L., Cherkassky, V.: Connection between SVM+ and multi-task learning. In: Proceedings of the International Joint Conference on Neural Networks, pp. 2048–2054 (2008)
20. Liu, Z., Jansen, B.J.: Factors influencing the response rate in social question and answering behavior. In: Proceedings of the 2013 Conference on Computer Supported Cooperative Work, pp. 1263–1274 (2013)
21. Rodrigues, T., Benevenuto, F., Cha, M., Gummadi, K., Almeida, V.: On word-of-mouth based discovery of the web. In: Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, pp. 381–396 (2011)
22. Romero, D.M., Galuba, W., Asur, S., Huberman, B.A.: Influence and passivity in social media. In: Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 18–33 (2011)
23. Stringhini, G., Wang, G., Egele, M., Kruegel, C., Vigna, G., Zheng, H., Zhao, B.Y.: Follow the green: growth and dynamics in Twitter follower markets. In: Proceedings of the 2013 Conference on Internet Measurement Conference, pp. 163–176 (2013)

24. Suh, B., Hong, L., Pirolli, P., Chi, E.H.: Want to be retweeted? Large scale analytics on factors impacting retweet in Twitter network. In: Proceedings of the 2010 IEEE Second International Conference on Social Computing, pp. 177–184 (2010)
25. Szabo, G., Huberman, B.A.: Predicting the popularity of online content. Commun. ACM **53**, 80–88 (2010)
26. Thomas, K., Grier, C., Song, D., Paxson, V.: Suspended accounts in retrospect: an analysis of Twitter spam. In: Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, pp. 243–258 (2011)
27. Vasconcelos, M., Almeida, J.M., Goncalves, M.A.: Predicting the popularity of micro-reviews: a foursquare case study. Inf. Sci. **325**, 355–374 (2015)
28. Zhao, Q., Erdogdu, M.A., He, H.Y., Rajaraman, A., Leskovec, J.: SEISMIC: a self-exciting point process model for predicting tweet popularity. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1513–1522 (2015)

Analysis of Teens' Chronic Stress on Micro-blog

Yuanyuan Xue[1,2](✉), Qi Li[1], Liang Zhao[1], Jia Jia[1], Ling Feng[1], Feng Yu[3], and David A. Clifton[4]

[1] Department of Computer Science and Technology,
Tsinghua University, Beijing, China
{xue-yy12,liqi13,jing-zhao11}@mails.tsinghua.edu.cn,
{jiajia,fengling}@mail.tsinghua.edu.cn

[2] Department of Computer Technology and Application,
Qinghai University, Xining, China

[3] Department of Psychology, Tsinghua University, Beijing, China
yufeng10@sem.tsinghua.edu.cn

[4] Institute of Biomedical Engineering, University of Oxford, Oxford, UK
davidc@robots.ox.ac.uk

Abstract. Statistics show that more and more teenagers today are under the stress in all areas of their lives from school to friend, work, and family, and they are not always able to use healthy methods to cope with. Long-term stress without proper guidance will lead to a series of potential problems including physical and mental disorders, and even suicide due to teens' shortage of psychological endurance and controllability. Therefore, it is necessary and important to sense teens' long-term stress and help them release the stress properly before the stress starts to cause illness. In this paper, we present a micro-blog based method to recognize teens' chronic stress by aggregating stress detected from micro-blog. In particular, we analyze the characteristics of teens' chronic stress, and identify five types of chronic stress level change patterns. We evaluate the framework through a user study at a high school where the 48 participants are aged 16–17. The result provides the evidence that sensing teens' chronic stress is feasible through the open micro-blog, and the identified stress level change patterns allow us to find useful regulations of teens' stress transition and to give sensible interpretations.

Keywords: Teens · Chronic pressure · Stress transition · Micro-blog

1 Introduction

1.1 Motivation

No one lives a stress-free life. Anything that poses a challenge or a threat to our well-being is a stress. The American Heritage Medical Dictionary defines *stress* as a physical or psychological stimulus that can produce mental or physiological reactions which may lead to illness [1]. Stress can be divided into *acute stress* or *chronic stress*. Acute stress is usually short-lived and can be beneficial, as it can

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 121–136, 2016.
DOI: 10.1007/978-3-319-48743-4_10

enhance alertness and improve productivity [2]. In contrast, chronic stress is long-lived. It is the response to emotional pressure suffered for a prolonged period over which an individual perceives s/he has no control. While the immediate effects of stress hormones are beneficial in a particular situation, long-term exposure to stress creates a high level of these hormones that remains constant. This may lead to high blood pressure and subsequently heart disease, damage to muscle tissue, inhibition of growth, suppression of the immune system, and damage to mental health [3]. In view of the severe consequence of chronic stress, it is quite important to catch it in time.

With the rapid economic development, chronic stress has become an epidemic in our modern society, and people almost accept it as a way of life. Particularly for teenagers, they have to face heightened stress due to the many changes experienced concomitantly. When teens are overloaded with long-term chronic stress, inadequately managed stress can lead to anxiety, withdrawal, aggression, physical illness, or poor coping skills such as drug/alcohol use. It could also trigger or worsen depression [4], social isolation, and aggressive miss-behaviors [5]. To the extreme, injury to either teenagers themselves or others will happen. The campus gunman, Elliot Rodger, posted a video of himself on YouTube, saying he had been suffering long-lasting stress of loneliness before the campus killing in Santa Barbara in USA [6].

Hence, being aware of the existence of chronic stress and helping stressful teenagers control and manage chronic stress before it becomes severe enough to cause illness are particularly important.

1.2 Existing Solutions

Traditional stress analysis and detection techniques use subjective questionnaires or various objective sensors to monitor the changes and predict the trends of *physiological signals* and/or *physical signals* for people under stress [7–9]. *Smart phones* as a kind of speech sensor were also exploited in [10,11], focusing on cognitive stress and stressor frequency estimation rather than the type and severity of stress. The limitations of these methods are the invasion or inconvenience caused by the body contact and the deviation induced by physical excise. Recently, micro-blog offers another low-cost sensing channel to obtain people's self-expressed contents and behaviors, from which some emotional signals could be captured and analyzed. [12–16] evaluated whether people are in the risk of depression by analyzing their twitting behaviors. Oriented at the youth group, [17–19] investigated a number of teens' typical tweeting behaviors that may reveal adolescent stress, and built a micro-blog based platform to sense and help ease teenagers' mental stress. But, aforementioned analysis stopped at detecting adolescent stress category and stress level from teenagers' tweets. None investigates further to design an approach to differentiate whether a teen suffers chronic stress upon the aggregation of stress of tweets within time periods on micro-blog.

1.3 Our Work

The aim of this study is to analyze and distinguish teenagers' acute and chronic stress through the social media micro-blog. Chronic stress could appear continuously or discontinuously and its level may vary over time during the stress period. Understanding teens' chronic stress level change patterns and features could help us track and predict teens' stress trend, then further provide proper intervention to avoid possible severe consequences.

Here, we turn to micro-blog for teens' chronic stress detection for the following reasons. Firstly, micro-blog keeps track of teens' long-term tweeting contents and behaviors, and it is possible to detect and associate stress within different time scopes, which is difficult for wearing body-contact sensors and taking the measurements. Secondly, teenagers tend to record details of their daily life and feelings on micro-blog, making the acquisition of teens' emotional states possible. Furthermore, based on the sensing results, effective and prompt intervention and interaction with teens under stress can be easily implemented through the lively micro-blog channel.

The contributions of this paper can be summarized as follows.

- By aggregating teens' stress levels from individual tweets, we design a method to differentiate teen's chronic stress over a time period.
- We analyze the characteristics of teens' chronic stress, and identify five types of chronic stress change patterns.
- We conduct a user study testing the accuracy of the proposed method with teenagers recruited from a local school, and experimentally analyze the reasons for different chronic stress level change patterns.

To our knowledge, this is the first attempt in the literature to analyze and identify teenagers' chronic stress, as well as different chronic stress level change patterns, on micro-blog social media.

2 Related Work

2.1 Stress Detection by Subjective Questionnaires and Psychologists

The first method uses subjective questionnaires or individual/group meetings with psychologists to analyze users' stress situations. This method needs high cooperation of users and relies on people's ability to recall their experiences.

2.2 Stress Detection from Physiological Signals

Because stress will induce the variation of physiological and physical signals of the body, there is a rich body of work using objective physiological and physical signals to detect stress. Typical physiological measures include galvanic skin response (GSR), heat rate variability (HRV), electroencephalogram

(EEG), electrocardiogram (ECG), blood pressure, electromyogram, and respiration. [20] found skin conductivity and heart rate metrics have close correlations with driver's stress level, and used electrocardiogram, electromyogram, skin conductance, and respiration for driver's stress detection [21]. [22] applied the non-linear system identification technique to HRV for continuous mental stress monitoring. The above experimental results came from the laboratories where subjects were under a stationary state. For people on the move, [23] combined ECG, GSR, and accelerometer gathered from 20 participants across three activities (sitting, standing, and walking) to differentiate physiological signals generated between physical activity and mental stress. [24] investigated the differences of EEG characteristics (overall complexity and spectrum power of EEG bands) collected from two groups of people - high stress versus moderate stress. The results showed that those with chronic stress have higher left prefrontal power.

2.3 Stress Detection from Physical Signals

Although physiological measures can achieve good accuracy in mental stress detection, it may make people discomfort, slightly conflicting air, or even more stressful increase due to its invasiveness. Some physical measures (e.g., voice, gesture and interaction, facial expressions, eye gaze, pupil dilation and blink rates) are thus taken for stress detection as non-invasive measures, since they do not need to put the contact sensors on human bodies. Voice-based stress detection has received much attention in recent years due to its observable variability when response to stressors. [25] presented a stress detection method by computing prosodic, voice quality, and spectral features on variable window sizes. Smart phones were also used as a kind of sensor to detect people's mental stress by analyzing their voice variation in diverse conversational situations [10,11]. Besides voice, [26] analyzed people's mouse movements and found that people click mouse button harder as their stress decrease. [27] proposed a stress recognition method based on pupil videos obtained from video camera. Considering the contact sensors need specialists to install and monitor, which causes inconvenience to people's daily life, [28] used a low-cost webcam that recovered the instantaneous heart rate signal from video frames of human faces for mental stress detection. [29] further used features extracted from GSR and/or speech signals to train four types of classifiers, and the result showed that SVM classifiers can reach better accuracy than other classifiers for stress detection.

2.4 Stress and Depression Detection from Micro-blog

The popularity of micro-blog offers another medium for sensing people's mood. [13,14] built a statistical classifier to estimate whether people are in the risk of depression by analyzing their twitting behaviors before being diagnosed. The experimental results demonstrated that social media contained useful cues in predicting individual's depression tendency. Recently, [17] investigated a number of teens' typical tweeting behaviors that may reveal adolescent stress, and applied five classifiers to teens' stress detection. [18] trained a deep sparse neural network to detect psychological stress from cross-media micro-blog. So far, teen's stress detection is conducted on the basis of individual tweets. Aggregation of stress levels revealed from tweets over a time period to identify teen's chronic stress and different chronic stress change patterns, have not been investigated yet. In this paper, a user study recruits 48 teenagers from a local high school and estimates the accuracy of the proposed method. Besides, some interesting findings are also found through the user study.

3 Problem Statement

Considering the characteristics of teenagers' micro-blog behaviors, [17,18] developed techniques to sense teenagers' stress level from each individual tweet of the four major categories: study, self-cognition, inter-personal, and affection. Six ranks: *none, very light, light, moderate, strong, very strong* are adopted to measure stress levels, where *none* indicates no stress. For computation purpose, we use integer set $\mathcal{S} = \{0, 1, 2, 3, 4, 5\}$ to represent the above labels. Let (t, w) be a tweet w posted at time t. Function $Stress(t, w) = s$ returns a detected stress level $s \in \mathcal{S}$ from tweet w.

To further sense chronic stress, we investigate a sequence of teen's tweets during a time period and aggregate the detected stress level of each single tweets.

Definition 1. *Let $I = [I.s, I.e]$ be a time period I starting time $I.s$ and ending time $I.e$. The temporal length of I is $|I| = I.e - I.s$, which can be a day, a week, a month, etc. Let $W(I) = \langle (t_1, w_1), (t_2, w_2) \cdots, (t_m, w_m) \rangle$ (for $I.s \leq t_1 \leq \cdots \leq t_m \leq I.e$) denote a* ***tweet sequence within time period I****, where tweets $w_1, w_2, \cdots, w_m$ were posted by a teenager chronologically on micro-blog at time $t_1, \cdots, t_m$, respectively.*

Definition 2. *Applying the stress detection function $Stress(t,\ w)$ upon each tweet in $W(I)$, we can obtain a corresponding* ***stress level sequence in time period I****, denoted as $S(W(I)) = \langle Stress(t_1, w_1), Stress(t_2, w_2), \cdots, Stress(t_m, w_m) \rangle = \langle s_1, s_2, \cdots, s_m \rangle$, where for $\forall i$ $(1 \leq i \leq m)$ $(s_i \in \mathcal{S})$. I is called a* ***stress existing time period****, if and only if $\exists i (1 \leq i \leq m)$ $(s_i > 0)$.*

As chronic stress is only meaningful for a relatively long time interval, and should exist frequently across the whole period, we give the following definition for chronic stress.

Definition 3. *Let $\mathcal{I} = [I_1, I_2, \cdots, I_n]$ be a time interval, which is divided into n successive time periods of equal temporal length, where for $\forall i$ ($1 \leq i \leq n-1$) ($I_i.e = I_{i+1}.s$) $\wedge$ ($|I_1| = |I_2| = \cdots = |I_n|$). For a list of tweet sequences posted within $\mathcal{I}$, $\mathcal{W} = [W(I_1), W(I_2), \cdots, W(I_n)]$, and a list of stress level sequences within $\mathcal{I}$, $\mathcal{S}(\mathcal{W}) = [S(W(I_1)), S(W(I_2)), \cdots,\ S(W(I_n))]$, the* ***stress coverage ratio within $\mathcal{L}$*** *is computed as the number of stress-existing time periods in $\mathcal{I}$ versus the total time periods number n.*

Assume $\mathcal{I}$ is a stress existing time period. $\mathcal{I}$ is called a ***chronic stress existing time period****, if and only if (1) the temporal length of $\mathcal{L}$ is greater than threshold τ_t, and (2) the stress coverage ratio within $\mathcal{L}$ is greater than a threshold τ_c. Based on teenagers' regular schedule, $\tau_t = 1$ month, and $\tau_c = 100\,\%$ for simplification of basic model in this study.*

4 Method

4.1 Gaussian Process for Single Tweet Stress Detection

We extract 9 features from teens' micro-blogs to characterize the postings related to stress. The features can be categorized into two types: content-centric (i.e., linguistic content, number of negative emotion words, shared music/picture genres, number of positive and negative emoticons, number of exclamations and question marks, emotional degree lexicons) and context-centric (abnormal tweeting time and frequency). From each teen's tweeting/retweeting behavior, we extract and analyze these features, and then employ a Gaussian Process classifier to perform single-tweet based stress detection. Several of these features are motivated from [17], where greater details can be accessed by the readers.

Based on the content and context features extracted from teens' tweets, we employ the Gaussian Process (GP) framework to learn the stress level (categorized into 6 levels: "No Stress", "Very Light", "Light", "Moderate", "Strong", "Very Strong") for each tweet, which offers a principled means of performing inference over noisy data. The significant reasons we adopt Gaussian Process are the notion of GP as a distribution over functions and its best performance for stress detection on micro-blog [17], thus it is suitable to analyze teens' tweets. Here, we still use 6 stress levels defined in previous work to measure individual's stress extent in single tweet. Detailed derivations can be found in [17].

4.2 Stress Aggregation on Single Tweets

Considering teens' routines of study and rest vary weekly, we set the granularity of time interval as "week", namely for the successive time interval $\mathcal{I} = [I_1, I_2, \cdots, I_n, I_m]$, the length $|I_i| (1 \leq i \leq m)$ is a week. Thus, for each teen, we first aggregate stress of single tweets weekly, using three typical functions Avg, Max, Sum to calculate the average, minimal, and maximal stress levels in a week. The aggregated stress value indicates the stress state of each week (whether the teen endures stress in this week or not).

Let $W(I) = \langle (t_1, w_1), (t_2, w_2) \cdots, (t_m, w_m) \rangle$ be a tweet sequence in time period I, and let $S(W(I)) = \langle Stress(t_1, s_1), Stress(t_2, s_2), \cdots, Stress(t_m, s_m) \rangle = \langle (t_1, s_1), (t_2, s_2), \cdots, (t_m, s_m) \rangle$ be a list of stress level sequences detected from $W(I)$ (*Definition 1* and *2*). We have $Avg(S(W(I))) = \frac{\sum_{i=1}^{m} s_i}{m}$, $Max(S(W(I))) = arg_{1 \leq i \leq m}\ max(s_i)$, and $Sum(S(W(I))) = arg_{1 \leq i \leq m}\ min(s_i)$.

We label the aggregation stress result (by $Avg/Max/Sum$ functions) in $|I_i| (1 \leq i \leq m)$ as $\mathcal{S}(W(I_i))$. Thus for a stress level sequence $S(\mathcal{W})$ in time interval $\mathcal{I}$, the aggregation result is $\mathcal{S}(\mathcal{W})$. In our later results presentation, we proved that teens' stress (aggregated weekly) has consistent changing trends under three aggregation methods.

4.3 Detecting Chronic Stress Level Change Patterns

Teen's stress state usually changes over time influenced by environment and personality during the chronic stress interval (for example, stress level rises fast or drops slowly). The changing of teens' stress reflects in the variation between low level and high level stress. Thus, in this paper, we define two stress states: *lower stress* and *higher stress*, and focus on their mutual transition.

Within the continuous chronic stress interval $\mathcal{I} = [I_1, I_2, \cdots, I_m]$ (where the length of $|I_i|(1 \leq i \leq m)$ is one week, $S(W(I_i)) > 0$), a teen has two different stress states *lower stress* and *higher stress*, measured by the stress value threshold τ. For the aggregated stress value $\mathcal{S}(W(I_i))$ of each time period $I_i(1 \leq i \leq m)$, if $\mathcal{S}(W(I_i)) \leq \tau$, I_i is in *lower stress* state; or else I_i is in *higher stress* state. The setting of τ is subject to teen's personality and daily behaviors (here we set τ to be 5 based on the fact of 5 working days in a week).

We define *lower stress* interval as $\mathcal{I}_l = [I_p, I_{p+1}, \cdots, I_k]$, and the adjacent *higher stress* interval as $\mathcal{I}_h = [I_{k+1}, I_{k+2}, \cdots, I_q]$ in time interval $\mathcal{I}$ $(1 \leq p \leq k \leq q \leq m)$, satisfying the following two conditions:

Condition 1: For $\forall i$ (p $\leq i \leq k$), $\mathcal{S}(W(I_i)) \leq \tau$;

Condition 2: For $\forall i$ (k+1 $\leq i \leq q$), $\mathcal{S}(W(I_i)) > \tau$.

Teen's chronic stress state changes from $\mathcal{I}_l$ to $\mathcal{I}_h$ when stress starts worsening. Similarly, transition from *lower stress* to *higher stress* happens when time interval $\mathcal{I}_h$ is in front of $\mathcal{I}_l$ in time line.

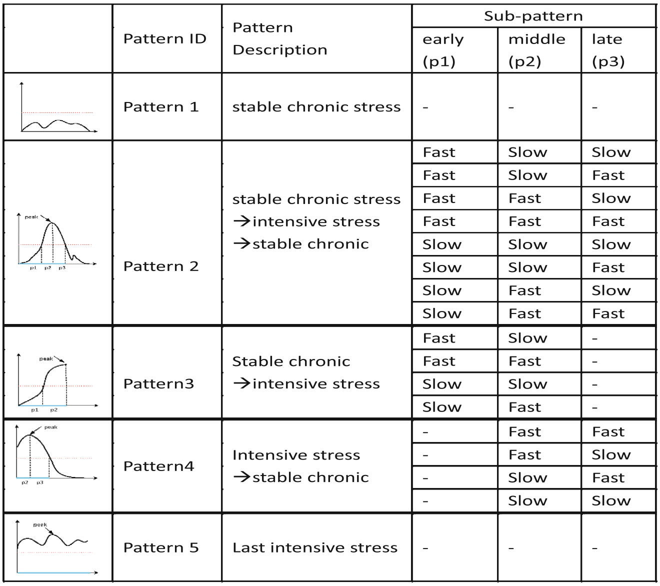
Within the *higher stress* time interval $\mathcal{I}_h$, we find the maximal stress level $\mathcal{S}_{peak}(I_h) = S(W(I_{peak}))$, where $S(W(I_{peak})) = arg_{p \leq i \leq k}\ Max(S(W(I_i)))$ $(p \leq i \leq k)$. I_{peak} (the time interval with peak stress value), together with I_p (the start time of $\mathcal{I}_l$) and I_q (the end time of $\mathcal{I}_h$), and divide the stress state transaction from *lower stress* to *higher stress* into three phases in the time line:

Phase 1. Early-Starting Phases stress level increases from $lowerstress$ state to $higherstress$ state, from I_p to I_k, in time span $T_1 = k - p$ (indicated in Fig. 1 with $p1$).

Phase 2. Middle-Rising Phases stress level increases from $S(W(I_k))$ to peak value $\mathcal{S}_{peak}(\mathcal{I}_h)$, from I_{k+1} to I_{peak}, in time $T_2 = peak - (k + 1)$ (indicated in Fig. 1 with $p2$).

Phase 3. Late-Decreasing Phases stress level decreases from peak value $\mathcal{S}_{peak}(\mathcal{I}_h)$ to *lower stress* state, from I_{peak} to I_q, in time $T_3 = q - peak$ (indicated in Fig. 1 with $p3$).

Note that the transition between *lower stress* and *higher stress* may occur repeatedly and alternately within a continuous chronic stress interval in reality. We measure the stress level changing speed (in *Phase2*) with the slope from I_k


peak
p1
p2
p3

	Pattern ID	Pattern Description	Sub-pattern		
			early (p1)	middle (p2)	late (p3)
	Pattern 1	stable chronic stress	-	-	-
	Pattern 2	stable chronic stress →intensive stress →stable chronic	Fast	Slow	Slow
			Fast	Slow	Fast
			Fast	Fast	Slow
			Fast	Fast	Fast
			Slow	Slow	Slow
			Slow	Slow	Fast
			Slow	Fast	Slow
			Slow	Fast	Fast
	Pattern3	Stable chronic →intensive stress	Fast	Slow	-
			Fast	Fast	-
			Slow	Slow	-
			Slow	Fast	-
	Pattern4	Intensive stress →stable chronic	-	Fast	Fast
			-	Fast	Slow
			-	Slow	Fast
			-	Slow	Slow
	Pattern 5	Last intensive stress	-	-	-

Fig. 1. Five chronic stress level change patterns.

to I_{peak}, denoted as $Speed_{up} = (\mathcal{S}(W(I_{peak})) - \mathcal{S}(W(I_k)))/T_2$. We measure the speed of *Phase1* and *Phase2* by using the time span T_1 and T_2. In our case study, we found 287 stress level transaction cases from 48 teens' tweets, and further calculated the average changing speed of three phases respectively from the 287 transactions, thus obtaining three thresholds for measuring the changing speed of each phase, which are denoted as λ_1, λ_2, λ_3.

Further, we present five stress level change patterns based on the transaction between *lower stress* and *higher stress* states, within the chronic stress interval $\mathcal{I} = [I_1, I_2, \cdots, I_m]$. For each pattern, sub-patterns are defined according to the changing speed ($fast$ or *slow*) of three phases, as shown in Fig. 1.

- Pattern 1: within the chronic stress interval $\mathcal{I}$, if for $\forall i$ $(1 \leq i \leq m)$, $\mathcal{S}(W(I_i)) \leq \tau$, we call $\mathcal{I}$ a *smooth stress pattern*, indicating that no *higher stress* state appears in this chronic stress interval.
- Pattern 2: within the chronic stress interval $\mathcal{I}$, if there exists three conjoint sub time intervals, $\mathcal{I}_l$, $\mathcal{I}_h$, $\mathcal{I}_l$ in time line, namely stress state changes from *lower stress* to *higher stress* state, then back to *lower stress* state, we call it a *burst stress pattern.*
- Pattern 3: within the chronic stress interval $\mathcal{I}$, if there exists two conjoint sub time intervals, $\mathcal{I}_l$ and $\mathcal{I}_h$ in time line, namely stress state changes from *lower stress* to *higher stress* state and lasts to the end of $\mathcal{I}$, we call it a *gradually intensified and long-lasting pattern.*
- Pattern 4: within the chronic stress interval $\mathcal{I}$, if there exists two conjoint sub time intervals, $\mathcal{I}_h$ and $\mathcal{I}_l$ in time line, indicating that stress changes from

higher stress (at the beginning of $\mathcal{I}$) to *lower stress* state, we call it a *downward stress pattern.*

- Pattern 5: within the chronic stress interval $\mathcal{I}$, if $\forall i$ $(1 \leq i \leq m)$, $\mathcal{S}(W(I_i)) \geq \tau$, we call $\mathcal{I}$ *intensive stress pattern*, showing that the teen keeps in *higher stress* state.

5 User Study

5.1 PSS-14 Questionnaires

Cohen's Perceived Stress Scale (PSS-14) [30] is commonly used to measure human's stress level worldwide in psychology. We take the PSS-14 score value (from 0 to 75, corresponding to none, light, moderate, and strong stress level respectively) as the ground-truth of our general stress detection results.

We invited 48 students (26 girls and 22 boys, aged 16–17) with micro-blog accounts from Xining No. 4 High School to participate in our case study. Getting their consents, we guided the students to fill in the Chinese PSS questionnaire [31] based on their emotional feelings in the last month (from `Nov.26,2014` to `Dec.26,2014`).

Besides, we also add two questions to ask participants to label their characters and micro-blog usage: (1) "*Do you think you are introvert or extrovert?*" and (2) "*Do you prone to express emotions through tweets?*"

5.2 Teens' Tweets from Tencent Micro-blog Platform

We collected 30,041 tweets before 26 December, 2014 of the above 48 students from Tencent Micro-blog Platform, which is similar to Twitter in China. For each teen, the number of tweets ranges from 28 to 3,288, and the time span ranges from 25 weeks to 261 weeks. These tweets provide us with abundant information to detect chronic and acute stress of the participants, and to further analyze their chronic stress level change patterns.

To verify our detection results for chronic stress according to our ground-truth (based on PSS-14 questionnaires), we collected tweets of the 48 teens in the corresponding month (from November 26, 2014 to December 26, 2014). For each teen, the number of tweets ranges from 1 to 94, and the time span ranges from 1 week to 5 weeks.

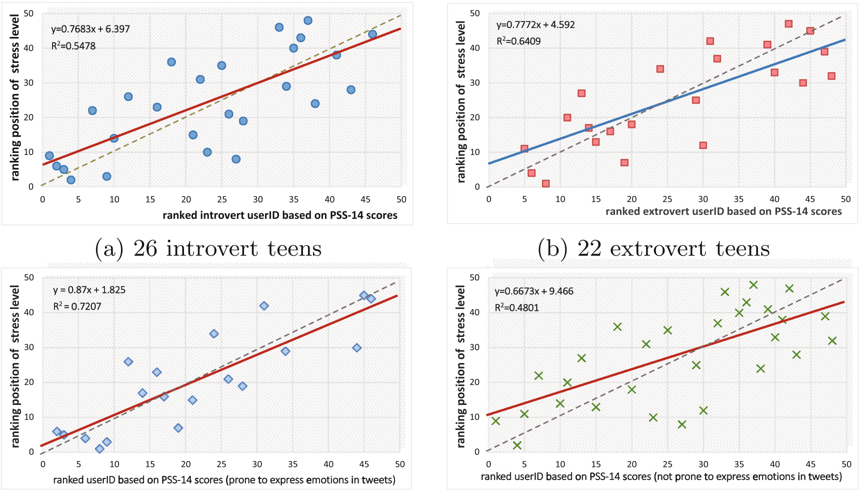
6 Results

To match the four stress levels of PSS-14, we merged our detected "*very light*" and "*light*" stress levels into "*light*" stress level, and "*very strong*" and "*strong*" stress levels into "*strong*" stress level.

6.1 Experiment 1: Teens' General Stress Detection

In this experiment, we used the subset of collected tweets (posted from November 26, 2014 to December 26, 2014), to guarantee the timeliness of our ground-truth (based on PSS-14 questionare). For each student, we detected the stress level of every single tweet, and further computed his/her average stress level in this month. By comparing the stress levels reflected by PSS-14 questionnaire and detected by our approach of each 48 students, the average detection accuracy of our approach is 77.1 %. 25 out of 29 students, who are identified as suffering strong stress by PSS-14, are detected correctly by our approach.

Effect of "Introvert" or "Extrovert" Character on Detection Performance. Figure 2(a) and (b) show the ranking results of PSS-14 and our approach based on the stress levels of 26 introvert students, and 22 extrovert students respectively. We draw the fitting straight line and use R^2 (ranging from 0 to 1) to show the linear correlation degree of data in each sub-figure. The R^2 of 22 extrovert students in Fig. 2(b) shows a higher value of 0.64 than that of 26 introvert students in Fig. 2(a). The difference of R^2 indicates that the ranking result of extrovert teens has a higher linear correlation degree than that of introvert teens.



(a) 26 introvert teens (b) 22 extrovert teens

(c) 19 teens prone to express emotions (d) 29 teens not prone to express emotions

Fig. 2. Ranking results comparison based on stress levels detected by our approach and PSS-14. We rank the PSS-14 scores among 48 teens, and also rank the average stress level of them detected by our approach. We compare the two rank results, using the data fitting method, where the result R^2 ranging from 0 to 1. The greater value of R^2 indicates higher fitting degree of the two ranks.

Students Who Likely Express Emotions on Micro-blog Get Higher Performance. Figure 2(c) and (d) show the ranking results of PSS-14 and our approach based on the stress levels of 19 teens who are prone to expressing emotions in tweets and 29 teens who are not prone to expressing emotions in tweets. The fitting straight line in Fig. 2(c) is nearer to the $y = x$ line and obtains a very high value of $R^2 = 0.72$, which is much higher than $R^2 = 0.48$ in Fig. 2(d), and also higher than the results in Fig. 2(a) and (b).

6.2 Experiment 2: Chronic Stress Detection

In this part, we used whole set of teens' tweets to detect teens' chronic stress and to analyze its change patterns during entire posting time. Given the above definition, in this research, we focused on stress level rather than stress type while detecting chronic stress.

Aggregation Performance. For each teen, we aggregated stress level of single tweets weekly by using three methods: (1) the maximal stress level in a week, (2) the average stress level in a week, (3) the accumulated stress level in a week.

Sensibility of Chronic Stress Intervals. According to the psychological theory, time span of chronic stress could be recognized as "months to years" [32]. Combined with the application scenarios (per semester lasting for 4 months in Chinese high school), the length threshold for chronic stress interval (in Definition 3) is set to 1, 2, 3 and 4 months respectively to explore teens' chronic stress reactions with different baseline time.

Result in Fig. 3 shows that the number of teens suffering from chronic stress gradually decrease as baseline time extended. Combing detection results and teens' tweets, we have observed that nearly half of teens release the stress when semester ends or vacation starts. The rest of teens endure stress coming from families, peers, or bad self-regulation capabilities, which are less affected by outside

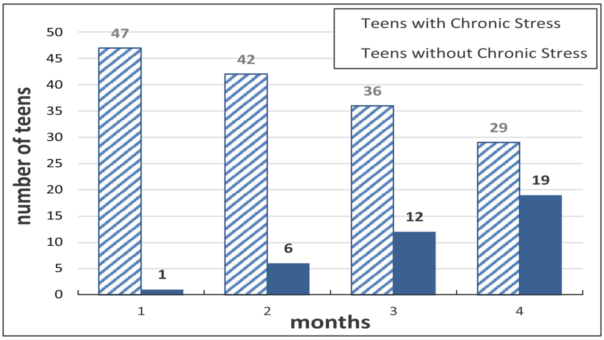


Fig. 3. Number of students with chronic stress, under different settings

world. When time span is set to 1 month, we find that most teens are suffering chronic stress with different periods. So we suggest that coaching or intervention once per month for teenagers be planned, to avoid potential consequences caused by chronic stress.

Performance of Chronic Stress Detection. To evaluate chronic stress detection performance, all participants were asked a question ("In the last year, do you feel nervous and stressed?" 1. Never 2. Seldom 3. Sometimes 4. Often 5. Always), which is regarded as ground-truth for detection result of each teen. To match question answers into the aggregation results (here we choose the average aggregation method, and the other two methods performing the similar trends can also be adopted), we set the rule: answer 1. Never mapping to "None", "2. Seldom" and "3. Sometimes" mapping to "Occasional", and "4. Often" and "5. Always" mapping to "Frequent". Here, we considered the label of "None" as no stress, "Occasional" as temporary stress, and "Frequent" as the chronic stress. Then we chose the minimal size of "1 month" as the threshold of chronic stress interval. The accuracy of detection result is presented in Table 1. Besides, the precision and recall of chronic stress detection can reach to 80.00 % and 66.67 % respectively. Causing relative low recall may be because some participants only posted few tweets during the posting period.

From another perspective, we compared our detection results with the rank of teens' PSS-14 scores (in descending order). Consequently, it reveals that most teens with chronic stress rank high in the list of PSS-14 results. As Table 2 shows, 10 teens with chronic stress are in the top 15, 13 teens are in the top 20, and 15 teens are in the top 25.

Table 1. Confusion matrix of stress detection

PSS-14	Detect		
	None	Occasional	Frequent (chronic)
None	1 (50 %)	1	0
Occasional	1	24 (77.42 %)	6
Frequent (chronic)	0	3	12 (80 %)

Table 2. Top n(15–25) ranks of teens with chronic stress

PSS-14 Rank	Top 15	Top 20	Top 25
Number of teens	10	13	15
Percent	66.67 %	86.66 %	100 %

6.3 Experiment 3: Chronic Stress Change Patterns

According to Definition 5, we explored all the varying patterns existing in the continuous chronic stress intervals of 48 teens. What we concerned more were the duration of low-level stress before its worsening, and the speed of worsening and alleviating. Five patterns were subclassified further in this trial. To describe the properties mentioned above, three parameters were used, including the lasting time of lower stress state $Lastingtime_{low}$, the transition speed from lower state to higher state $Speed_{up}$, and the transition speed from higher state to lower state $Speed_{down}$. We marked $Lastingtime_{low}$ as *slow* when it was greater than average value of lasting time of 48 teens, and marked $Speed_{up}$ and $Speed_{down}$ as *slow* when they were less than average value, the other way round. In such cases, stress level change patterns were classified into 18 sub-patterns (in Definition 4).

Table 3 shows the comparison of occurrence proportion of 5 patterns when the baseline time is set to 1, 2, 3 and 4 months. In general, the overall proportion do not markedly change with baseline time variation. Among these patterns, pattern 1 is the dominant one and then is pattern 2, which means teens usually are under the lower stress while suffering from chronic stress. But when their stress intensifies, it usually turns into a way of increasing first and then decreasing. The small change of percentage of pattern 1 and pattern 2 indicate that the ratio of stress worsening will rise with baseline time increasing.

Table 3. Distribution of 5 chronic stress change patterns in different baseline time

	Pattern 1	Pattern 2	Pattern 3	Pattern 4	Pattern 5
1 month	75.04 %	15.91 %	3.47 %	4.86 %	0.69 %
2 month	74.32 %	17.96 %	3.50 %	3.85 %	0.35 %
3 month	72.06 %	20.37 %	3.39 %	4.01 %	0.15 %
4 month	67.15 %	25.24 %	3.18 %	4.16 %	0.24 %

The proportion of sub-patterns in pattern 2 in 1 month baseline time is shown in Fig. 4 (only one pattern is reported here due to the space restriction). It shows that an overwhelming majority of teens (87.37 %) turns to the stage of intensive stress only after a short period of lower stress, which might be because teens easily feel anxious when facing stress due to their immaturity. Besides, the most prominent sub-pattern is sub-pattern 4, in which the speed of both stress ascent and decline are "fast", which seems like "easy come easy go" when teens' stress get worse. But an important thing to know is there are still 45.6 % teens keep "slow" speed at stress alleviating stage. For these teens, intervention or psychology guidance should be imported early to help them release stress.

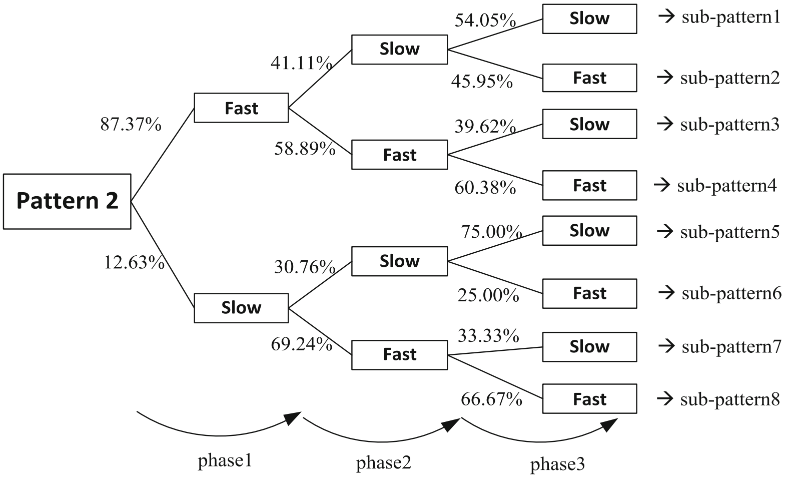


Fig. 4. Detected proportions of 8 sub-patterns in pattern 2

7 Conclusion

Chronic stress can lead to a series of physical and mental health problems. Especially for adolescents, it may result in more serious consequences such as suicide, due to their shortage of psychological endurance and controllability. Therefore, it is particularly necessary to timely detect adolescents' chronic stress and guide them to cope with it. In this study, we propose a framework for chronic stress detection by aggregating individual tweet's stress detection results. We identify five chronic stress change patterns, and give explanations and some advices based on the findings from a user study with 48 students of age 16–17 recruited from a high school. In the future, we plan to implant a personalization model upon the framework to automatically or semi-automatically adapt to different teens' stress detection.

Acknowledgments. The work is supported by National Natural Science Foundation of China (61373022 and 61370023).

References

1. The American Heritage Medical Dictionary. Houghton Mifflin Company (2008)
2. Benson, H., Allen, R.: How much stress is too much? Harvard Bus. Rev. **58**(5), 86–92 (1980)
3. Stress. http://psychology.wikia.com/wiki/Stress
4. Stress contributes to range of chronic diseases, review shows (2007). http://www.sciencedaily.com/releases/2007/10/071009164122.htm
5. Mineur, Y., Prasol, D., Belzung, C.: Agonistic behavior and unpredictable chronic mild stress in mice. Behav. Genet. **33**(5), 513–519 (2003)

6. Psychology today (2014). http://www.psychologytoday.com/blog/web-loneliness/201405/the-loneliness-elliot-rodger
7. Shi, Y., Ruiz, N., Taib, R., Choi, E., Chen, F.: Galvanic skin response (GSR) as an index of cognitive load. In: Proceedings of CHI, pp. 2651–2656 (2007)
8. Hamid, N., Sulaiman, N., Aris, S., Murat, Z., Taib, M.: Evaluation of human stress using EEG power spectrum. In: Proceedings of CSPA, pp. 1–4 (2010)
9. Hosseini, S., Khalilzadeh, M.: Emotional stress recognition system using EEG and psychophysiological signals: using new labelling process of EEG signals in emotional stress state. In: Proceedings of ICBECS, pp. 1–6 (2010)
10. Lu, H., Rabbi, M., Chittaranjan, G., Frauendorfer, D., et al.: Stresssense: detecting stress in unconstrained acoustic environments using smartphones. In: Proceedings of Ubicomp, pp. 351–360 (2012)
11. Bauer, G., Lukowicz, P.: Can smartphones detect stress-related changes in the behaviour of individuals? In: Proceedings of PERCOM Workshop, pp. 423–426 (2012)
12. Park, M., McDonald, D., Cha, M.: Perception differences between the depressed and non-depressed users in Twitter. In: Proceedings of ICWSM, pp. 476–485 (2013)
13. Choudhury, M., Gamon, M., Counts, S., Horvitz, E.: Prediction depression via social media. In: Proceedings of ICWSM, pp. 128–137 (2013)
14. Choudhury, M., Counts, S., Horvitz, E.: Social media as a measurement tool of depression in populations. In: Proceedings of ACM Web Science, pp. 47–56 (2013)
15. Shen, Y.-C., Kuo, T.-T., Yeh, I.-N., Chen, T.-T., Lin, S.-D.: Exploiting temporal information in a two-stage classification framework for content-based depression detection. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013. LNCS (LNAI), vol. 7818, pp. 276–288. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37453-1_23
16. Wang, X., Zhang, C., Ji, Y., Sun, L., Wu, L., Bao, Z.: A depression detection model based on sentiment analysis in micro-blog social network. In: Li, J., Cao, L., Wang, C., Tan, K.C., Liu, B., Pei, J., Tseng, V.S. (eds.) PAKDD 2013. LNCS (LNAI), vol. 7867, pp. 201–213. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40319-4_18
17. Xue, Y., Li, Q., Jin, L., Feng, L., Clifton, D.A., Clifford, G.D.: Detecting adolescent psychological pressures from micro-blog. In: Zhang, Y., Yao, G., He, J., Wang, L., Smalheiser, N.R., Yin, X. (eds.) HIS 2014. LNCS, vol. 8423, pp. 83–94. Springer, Heidelberg (2014). doi:10.1007/978-3-319-06269-3_10
18. Lin, H., Jia, J., Guo, Q., Xue, Y., Li, Q., Huang, J., Cai, L., Feng, L.: User-level psychological stress detection from social media using deep neural network. In: Proceedings of MM (2014)
19. Li, Q., Xue, Y., Jia, J., Feng, L.: Helping teenagers relieve psychological pressures: a micro-blog based system. In: Proceedings of EDBT Demo (2014)
20. Healey, J., Picard, R.: Detecting stress during real-world driving tasks using physiological sensors. IEEE Trans. Intell. Trans. Syst. **6**(2), 156–166 (2005)
21. Rigas, G., Goletsis, Y., Fotiadis, D.: Real-time driver's stress event detection. IEEE Trans. Intell. Trans. Syst. **13**(1), 221–234 (2012)
22. Choi, J., Gutierrez-Osuna, R.: Using heart rate monitors to detect mental stress. In: Wearable and Implantable Body Sensor Networks, pp. 219–223 (2009)
23. Sun, F., Kuo, C., Cheng, H., Buthpitiya, S., Collins, P., Griss, M.: Activity-aware mental stress detection using physiological sensors. In: Proceedings of Social Informatics and Telecommunications Engineering, pp. 211–230 (2012)

24. Peng, H., Hu, B., Zheng, F., Fan, D., Zhao, W., Chen, X., Yang, Y., Cai, Q.: A method of identifying chronic stress by EEG. Pers. Ubiquit. Comput. **17**(7), 1341–1347 (2013)
25. Soury, M., Devillers, L.: Stress detection from audio on multiple window analysis size in a public speaking task. In: Proceedings of Affective Computing and Intelligent Interaction, pp. 529–533 (2013)
26. Liao, W., Zhang, W., Zhu, Z., Ji, Q.: A real-time human stress monitoring system using dynamic Bayesian network. In: Proceedings of CVPR (2005)
27. Mokhayeri, F., Akbarzadeh-T, M.-R.: Mentail stress detection based on soft computing techniques. In: Bioinformatics and Biomedicine, pp. 430–433 (2011)
28. Bousefsaf, F., Maaoui, C., Pruski, A.: Remote assessment of the heart rate variability to detect mental stress. In: Proceedings of Pervasive Computing Technologies for Healthcare Workshops, pp. 348–351 (2013)
29. Kurniawan, H., Maslov, A., Pechenizkiy, M.: Stress detection from speech and galvanic skin response signals. In: Proceedings of CBMS, pp. 209–214 (2013)
30. Cohen, S., Kamarck, T., Mermelstein, R.: A global measure of perceived stress. J. Health Soc. Behav., 385–396 (1983)
31. Cheng, T., Wu, J., Chong, M., Williams, P.: Internal consistency and factor structure of the Chinese health questionnaire. Acta Psychiatr. Scand. **82**(4), 304–308 (1990)
32. Contrada, R.J.: The Handbook of Stress Science. Springer Publishing Company. LLC, New York (2011)

Large-Scale Stylistic Analysis of Formality in Academia and Social Media

Thin Nguyen(✉), Svetha Venkatesh, and Dinh Phung

Deakin University, Burwood, Australia
{thin.nguyen,svetha.venkatesh,dinh.phung}@deakin.edu.au

Abstract. The dictum 'publish or perish' has influenced the way scientists present research results as to get published, including exaggeration and overstatement of research findings. This behavior emerges patterns of using language in academia. For example, recently it has been found that the proportion of positive words has risen in the content of scientific articles over the last 40 years, which probably shows the tendency in scientists to exaggerate and overstate their research results. The practice may deviate from impersonal and formal style of academic writing. In this study the degree of formality in scientific articles is investigated through a corpus of 14 million PubMed abstracts. Three aspects of stylistic features are explored: expressing emotional information, using first person pronouns to refer to the authors, and mixing English varieties. Trends of these stylistic features in scientific publications for the last four decades were discovered. A comparison on the emotional information with other online user-generated media, including online encyclopedias, web-logs, forums, and micro-blogs, was conducted. Advances in cluster computing are employed to process large scale data, with 5.8 terabytes and 3.6 billions of data points from all the media. The results suggest the potential of pattern recognition in data at scale.

Keywords: Big data · Apache Spark · Stylistic features · Academia · Social media

1 Introduction

Publication pressure has influenced the way researchers present study results, emerging patterns of presentations in scientific publications. Recently a pattern in expressing emotional information in academic papers has been found: the proportion of positive words has risen in the content of academic articles for the last 40 years [2,13], linking with the possibility of exaggeration or distortion of study findings [7,13], in order to give the papers more chance to be accepted. As emotional expression is said to be avoided in academic writing [1] or to associate with informal writing [11], the writing mode may not follow a formal style.

Other informal elements include using first person pronouns to refer to the authors [4,5]. Academic writers are advised that '*leave their personalities at the door*' [6] and '*traditional formal writing does not use I or we in the body of the paper*' [12].

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 137–145, 2016.
DOI: 10.1007/978-3-319-48743-4_11

Another requirement for scientific publications is the consistency in the use of English. In the guide to authors by several journals, such as *Journal of Phonetics*[1] or *Information Sciences*,[2] it is written '*Please write your text in good English (American or British usage is accepted, but not a mixture of these)*'.

This work adapts a data-driven and lexicon-based approach to capture the language style conveyed in PubMed articles. Not only two linguistic categories – positives and negatives, but other stylistic features expressed in the content of PubMed abstracts will also be extracted. They include the proportion of sentiment-bearing words in the content and their affective scores, the extent of using first person pronouns to refer to the authors, and the degree of mixing English spelling in academic writing. Trends of these stylistic features in scientific publications for the last four decades will be investigated. The emotional information will be compared with that of other media, consisting of online encyclopedia (Wikipedia), online diaries (web-logs, e.g., Live Journal), online forums (Reddit), and micro-blogs (Twitter). Advanced framework in cluster computing will be employed to process approximately six terabytes of data containing billions of data points from all the media.

A key contribution of this work is to provide a set of stylistic features for scientific articles as well as trends of these features for the last 40 years. This would probably help to understand the evolution of scientific writing, as well as anomalies along the development. The trends may also imply changes in the extent of acceptability of certain stylistic features over the course of the period.

The paper is organized as follows. Section 2 outlines the proposed methods, data collections, and experimental setup. Section 3 presents the results. Section 4 concludes the paper and proposes possible future work.

2 Method

2.1 Datasets

Data from PubMed, Wikipedia, Live Journal, Reddit, and Twitter were crawled or downloaded. The time range, number of instances, and the volume for these corpora is shown in Table 1.

For PubMed, on 20 January 2016, through http://www.ncbi.nlm.nih.gov/pubmed/, English articles having abstracts were queried. The site returned 14,334,783 records of articles, in XML format. For each record, the content of certain tags, such as article title, article abstract, and publication date, was extracted.

For Wikipedia, the 05 March 2016 dump of English Wikipedia was downloaded.[3] This contains 8,374,298 articles. However, many of them do not have

[1] https://www.elsevier.com/journals/journal-of-phonetics/0095-4470/guide-for-authors, 2016.

[2] https://www.elsevier.com/journals/information-sciences/0020-0255/guide-for-authors, 2016.

[3] https://dumps.wikimedia.org/enwiki/20160305, downloaded on 31 March 2016.

Table 1. Corpora used in the experiments.

Dataset	Time range	#Instance	Volume
PubMed	01/12/1948 – 20/01/2016	14,334,783	150 GB
Wikipedia	05/11/2002 – 05/03/2016	5,760,798	52 GB
Live journal	14/05/1999 – 23/04/2005	33,152,794	64 GB
Reddit posts	24/01/2006 – 31/08/2015	196,531,736	251 GB
Reddit comments	15/10/2007 – 31/05/2015	1,659,361,605	908 GB
Twitter	07/06/2013 – 14/03/2016	1,673,497,746	4,542 GB

the content. They are referrals of other articles and were removed, resulting in 5,760,798 articles.

For Live Journal, a corpus of RSS (Rich Site Summary) feeds provided by the authors of [8] was used. This dataset contains more than 33 million blog posts written in English. More than half of the posts are tagged with moods, probably suggesting a sentiment-bearing and less formal data source.

For Reddit, both corpora of posts and comments were downloaded, including approximately 200 million posts and 1.6 billion comments.[4]

For Twitter, we introduce a new dataset of tweets written in English, geo-tagged with a US location, and time-stamped from 07 June 2013 to 14 March 2016.

2.2 Features

To characterize PubMed articles, three types of features were extracted: (1) affective information conveyed in the content, (2) informal elements, and (3) the mixing of American and British English. For other media, the affective information was extracted.

Affective Information. To compute the proportion of affective words in a document, *positive emotions* and *negative emotions* lexicons from LIWC [10] were employed. To estimate the sentiment score of sentiment-bearing words, ANEW ratings [3], employed to infer mood patterns in [9], was used. Words in this lexicon are rated in term of valence (*very unpleasant* to *very pleasant*), arousal (*least active* to *most active*), and dominance (*submissive* to *dominant*).

Using First Person Pronouns. In this work, using first person pronouns to refer to the authors was used as an example of informal elements. In particular, the proportion of abstracts containing *We* and *I* was calculated.

Generally, *We* is used more in co-authored papers than in sole-authored ones, and the other way around for *I*. So, in addition to the proportion calculated for all papers, those computed with respect to sole- and co-authored papers are included.

[4] Posts: http://bit.ly/1MvQobz, comments: http://bit.ly/1RmhQdJ, retrieved October 2015.

Mixing of American (AmE) and British English (BrE) Spelling. This work examined differences between AmE and BrE in term of spelling. From http://www.studyenglishtoday.net/british-american-spelling.html,[5] all words with different spelling between AmE and BrE were selected as the initial set. Then those words highlighted as misspelling by Notepad++ version 6.7.5 and their British counterparts were chosen, resulting in a list of 60 American and British word couples.[6]

This list was employed as the vocabulary to determine if a mixture of American and British spelling was used in a document of the media. For example, if both ‘behavior’ and ‘colour’ were found in a document, the document would be considered to have a mixture of English varieties in the content.

2.3 Trends in Academic Writing

To examine trends of stylistic features in scientific articles, annual values of each feature were examined. Correlation of the values with time would roughly show how the features have been changed over time. In this work, Pearson correlation between the annual value of the features and the time was used to detect the evolution of academic writing in PubMed articles for the last four decades, from 1975 to 2015. A positive correlation possibly means that the feature has increased over the time, and vice versa.

2.4 Computing Environment

All the processing of the big corpora in this work was conducted using Apache Spark, an emerging cluster computing platform [14]. A Spark cluster of eight worker nodes was employed. Each node features a dual eight-core Intel® Xeon® E5-2670@2.60 GHz processors, 128 GB of main memory, and CentOS 7.2 operating system.

3 Results

3.1 Affective Information

Proportion of Positive and Negative Words in the Content. Figure 1 shows the use of positive and negative words in PubMed abstracts for the last four decades. The use of positive words has increased for the period, with the correlation with time of 0.79, partly confirming recent findings that the proportion of positive words has risen in the content of scientific articles [2,13,15]. On the other hand, virtually no trend is observed for the use of negative words over the time.

[5] Retrieved February 2016, cached: http://bit.ly/1UNOeWa.

[6] The list could be accessed at http://bit.ly/1Sb1E9Z.

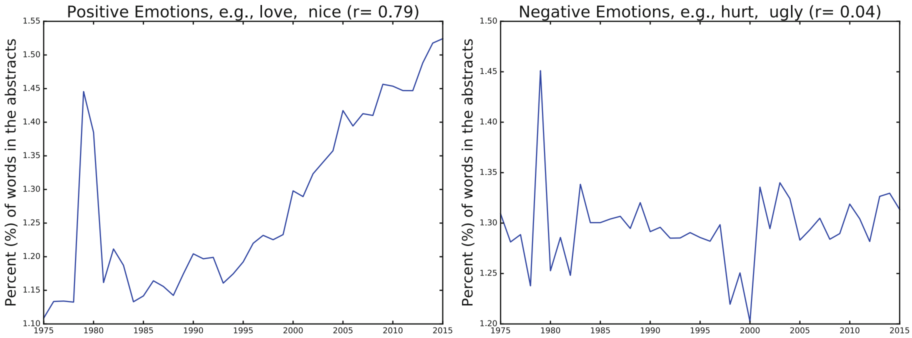
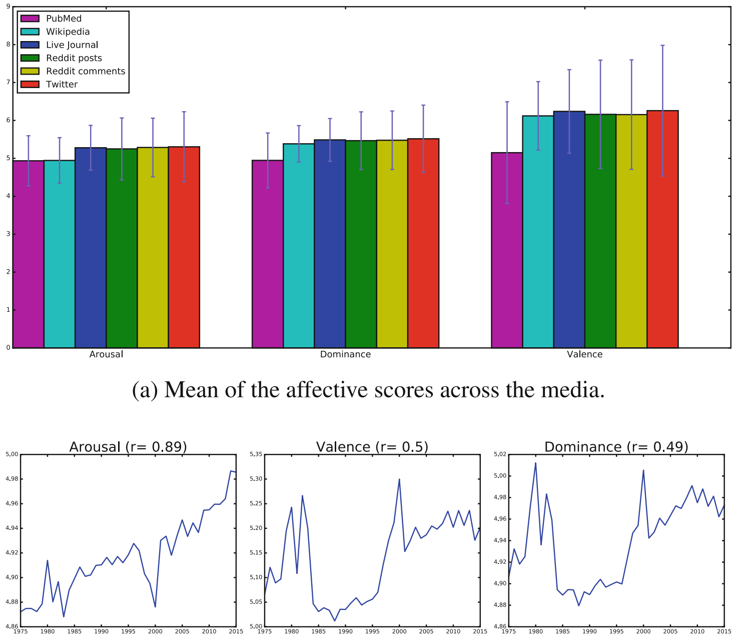


Fig. 1. The use of positive and negative words in PubMed abstracts for the last 40 years.



(a) Mean of the affective scores across the media.

(b) The affective scores conveyed in PubMed abstracts.

Fig. 2. The affective scores for all the media and the affective scores for PubMed articles for the last 40 years.

Affective Score of Subjective Words. Figure 2a shows the mean of three affective scores for the content of all the media. The value for PubMed abstracts is lowest in all the affective scores.

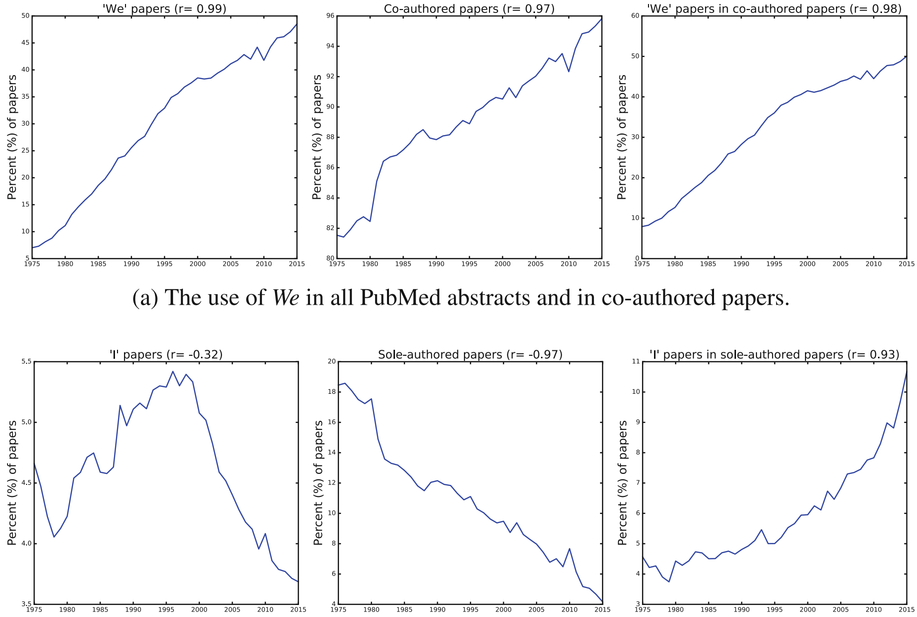
However, all the scores for PubMed abstracts have increased for the last 40 years, as shown in Fig. 2b. Especially, the trend is clear for *arousal* score, with strong correlation with time, at 0.89. So, it could be said that, for PubMed abstracts, an increase is seen not only in the proportion of sentiment-bearing words in the content (Fig. 1), but also in the scores of affective words (Fig. 2b).

3.2 Using First Person Pronouns

Figure 3 shows the use of one of the informal elements – using first person pronouns to refer to the authors – in PubMed publications for the last 40 years. In general, an increase is seen in the use of *We* and *I* in PubMed abstracts over the period.

In particular, for all abstracts, *We* was found to be increasingly used, from less than 10 % of papers in 1975 to almost 50 % in 2015, a five times higher, shown in the left figure of Fig. 3a. A similar trend is observed for the proportion of *We* papers (papers with *We* in the abstracts) in co-authored papers, shown in the right figure of Fig. 3a. An increase in the proportion of co-authored papers is also seen, from less than 82 % in 1975 to almost 96 % in 40 years later, shown in the middle figure of Fig. 3a.

On the other hand, for all abstracts, as shown in the left figure of Fig. 3b, the percentage of *I* papers (papers with *I* in the abstracts) is slightly decreased over the time. However, the reason for this decrease is due to the sharp drop



(a) The use of *We* in all PubMed abstracts and in co-authored papers.

(b) The use of *I* in all PubMed abstracts and in sole-authored papers.

Fig. 3. The use of first person pronouns in all PubMed abstracts, as well as with respect to co-authored or sole-authored papers, for the last 40 years.

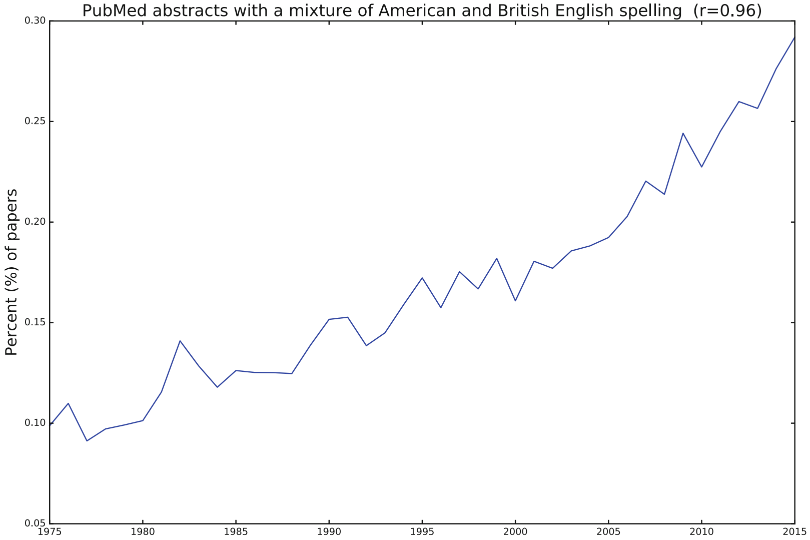


Fig. 4. Proportion of PubMed abstracts with a mixture of American and British English spelling over the last 40 years.

in the proportion of sole-authored papers, from more than 18 % in 1975 to 4 % in 2015, shown in the center figure of 3b. Indeed, the percentage of *I* papers in sole-authored papers has increased for the last 40 years, from 4.5 % to more than 10 %, a more than two times higher, shown in the right figure of Fig. 3b.

The increases in the proportions of *We* and *I* papers in co-authored and sole-authors, respectively, probably imply a rise in the degree of acceptability of the informal element in academic writing.

3.3 Mixing of American and British English Spelling

As shown in Fig. 4, the rate of PubMed articles containing a mixture of English spelling has increased for the last 40 years. In 1975, only 0.1 % of PubMed abstracts had a mixture of American and British English spelling. In 2015, this number is almost 0.3 %, a three times higher in 40 years.

4 Conclusion and Future Work

The work investigated stylistic features expressed in PubMed papers published for the last 40 years, with a comparison on the affective information with a variety of media, consisting of online encyclopedia, online diaries, online forums, and micro blogs. Advances in cluster computing framework were utilized to process almost 6 terabytes of data. Emerging trends in academic writing have been discovered. Among others, there exists the tendency of using first person pronouns to refer to the authors and mixing English spelling in the abstracts for the last four decades. Results also indicated differences in the affective information between academia and other media. The work demonstrated the efficiency of advanced computing framework in dealing with big data, providing prompt results.

The result is limited to publications in bio-medical and life sciences. Future studies should consider scientific articles in other fields to further validate the findings. Future research would also benefit from conducting sub-analyses for scientific publications, such as broken by journal impact factors and author affiliations (or English as the first or the second language for the authors). This would help to gain deeper insight into stylistic differences among sub-cohorts of academic articles.

Furthermore, other informal elements as well as other differences among the English varieties should be included in future work to capture a comprehensive view of academic writing.

Acknowledgment. This work is partially supported by the Telstra-Deakin Centre of Excellence in Big Data and Machine Learning.

References

1. Ahmad, J.: Stylistic features of scientific English: a study of scientific research articles. English Lang. Lit. Stud. **2**(1), 47 (2012)
2. Ball, P.: 'Novel, amazing, innovative': positive words on the rise in science papers. Nature (2015)
3. Bradley, M.M., Lang, P.J.: Affective norms for English words (ANEW): instruction manual and affective ratings (1999)
4. Burrough-Boenisch, J.: Negotiable acceptability: reflections on the interactions between language professionals in Europe and NNS 1 scientists wishing to publish in English. Curr. Issues Lan. Plan. **7**(1), 31–43 (2006)
5. Chang, Y.Y., Swales, J.: Informal elements in English academic writing: threats or opportunities for advanced non-native speakers. In: Writing: Texts, Processes and Practices, pp. 145–167. Longman (1999)
6. Hyland, K.: Options of identity in academic writing. ELT J. **56**(4), 351–358 (2002)
7. Lazarus, C., Haneef, R., Ravaud, P., Boutron, I.: Classification and prevalence of spin in abstracts of non-randomized studies evaluating an intervention. BMC Med. Res. Methodol. **15**(1), 1 (2015)
8. Leshed, G., Kaye, J.J.: Understanding how bloggers feel: recognizing affect in blog posts. In: Proceedings of Conference on Human Factors in Computing Systems, pp. 1019–1024 (2006)
9. Nguyen, T.: Mood patterns and affective lexicon access in weblogs. In: Proceedings of ACL Student Research Workshop, pp. 43–48 (2010)
10. Pennebaker, J.W., Boyd, R.L., Jordan, K., Blackburn, K.: The development and psychometric properties of LIWC2015. Technical report, University of Texas at Austin (2015)
11. Rosen, L.D., Chang, J., Erwin, L., Carrier, L.M., Cheever, N.A.: The relationship between 'textisms' and formal and informal writing among young adults. Commun. Res. **37**(3), 420–440 (2010)

12. Spencer, C.M., Arbon, B.: Foundations of Writing: Developing Research and Academic Writing Skills. National Textbook Company, Lincolnwood (1996)
13. Vinkers, C.H., Tijdink, J.K., Otte, W.M.: Use of positive and negative words in scientific PubMed abstracts between 1974 and 2014: retrospective analysis. BMJ **351**, h6467 (2015)
14. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I.: Spark: cluster computing with working sets. In: Proceedings of 2nd USENIX Conference on Hot Topics in Cloud Computing, p. 10 (2010)
15. Zimmer, C.: Staying afloat in the rising tide of science. Cell **164**(6), 1094–1096 (2016)

Discriminative Cues for Different Stages of Smoking Cessation in Online Community

Thin Nguyen[1(✉)], Ron Borland[2], John Yearwood[1], Hua-Hie Yong[2], Svetha Venkatesh[1], and Dinh Phung[1]

[1] Deakin University, Geelong, Australia
{thin.nguyen,john.yearwood,svetha.venkatesh,dinh.phung}@deakin.edu.au
[2] Cancer Council Victoria, Melbourne, Australia
{ron.borland,hua.yong}@cancervic.org.au

Abstract. Smoking is one of the leading causes of preventable death, being responsible for about six million deaths annually worldwide. Most smokers want to quit, but many find quitting difficult. The Internet enables people interested in quitting smoking to connect with others via online communities; however, the characteristics of these discussions are not well understood. This work aims to explore the textual cues of an online community interested in quitting smoking: www.reddit.com/r/stopsmoking – "*a place for redditors to motivate each other to quit smoking*". A total of approximately 5,000 posts were randomly selected from the community. Four subgroups of posts based on the cessation days of abstainers were defined: S0: within the first week, S1: within the first month (excluding cohort S0), S2: from second month to one year, and S3: beyond one year. Psycho-linguistic features and content topics were extracted from the posts and analysed. Machine learning techniques were used to discriminate the online conversations in the first week S0 from the other subgroups. Topics and psycho-linguistic features were found to be highly valid predictors of the subgroups, possibly providing an important step in understanding social media and its use in studies of smoking and other addictions in online settings.

Keywords: Feature extraction · Textual cues · Web community · Smoking cessation

1 Introduction

Internet is increasingly being used for the exchange of information, support, and advice on a range of health concerns, including dealing with certain addictions, such as, smoking, drinking, and drug abuse. Reddit is one such avenue for people who share a common interest to connect and form communities with a specific interest. Such communities are known as subreddits and their members are called redditors. Online users can contribute to these subreddits by making posts and getting them discussed and commented on. For dealing with addiction, people may join subreddits of their interest, e.g., "r/stopsmoking", "r/stopdrinking",

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 146–153, 2016.
DOI: 10.1007/978-3-319-48743-4_12

or "r/stopgaming". In these communities abstainers could exchange their own health story, encourage others, or record their journey of self-treatment, such as getting rid of smoking, drinking, or gaming. However, to date, little is known about the topics discussed within these communities or the language features that characterize these discussions.

This study aims to examine the topics and linguistic features in an online community interested in smoking cessation www.reddit.com/r/stopsmoking – "*a place for redditors to motivate each other to quit smoking.*" A large corpus of data was crawled including thousands of posts made by thousands of users within the community. We present an analysis focusing on the topics and psycholinguistic processes expressed in the content of users' posts, to identify predictive feature sets.

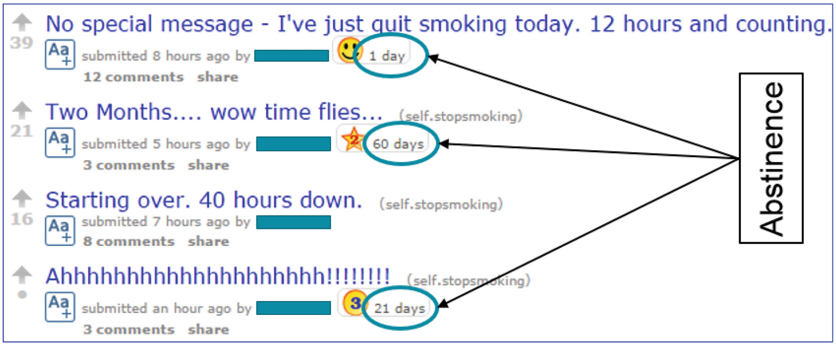
A key contribution of this work is to provide a comprehensive view based on topics of interest and language styles of members of an addiction community who self-identified as smokers who have quit smoking. Another contribution is to provide a set of predictors to differentiate users in different stages of smoking cessation. This work helps to improve our understanding of online addiction communities and illustrates the potential of machine learning for improving health care research and practice.

The current paper is organized as follows. Section 2 presents related work. Section 3 outlines the proposed methods and experimental setup. Section 4 presents the performance of topics and language styles in classifying posts into different stages of quitting. Section 5 concludes the paper and notes the prospect for future research.

2 Related Work

Several studies have considered Reddit as a new venue for exploration. For example, subreddits "r/stopsmoking" and "r/stopdrinking" were investigated to gain insight into smoking and drinking cessation [8]; "r/suicidewatch" was explored to discover changes in its content following celebrity suicides [4]; "r/stopsmoking", "r/hookah", and "r/electronic_cigarette" were examined to investigate into people's experiences with different tobacco products [2].

To conceptualize the content, two feature sets have been widely used: (1) topics: *what* people are writing about and (2) language styles: *the way* they express the story. To extract topics, latent Dirichlet allocation (LDA) [1] – a Bayesian probabilistic topic modeling – is often employed. To capture language styles conveyed in the content, packages proposed in psychology, such as LIWC [7], is widely used. For example, both topics and language styles were used as the base to detect community [5]. These two representations also potentially provide insights into mental health status of individuals. Indeed, these features have been found to be strong predictors of autism, and differentiate mental health communities from other online communities [6].



39 No special message - I've just quit smoking today. 12 hours and counting.
submitted 8 hours ago by [redacted] 1 day
12 comments share

21 Two Months.... wow time flies... (self.stopsmoking)
submitted 5 hours ago by [redacted] 2 60 days
3 comments share

16 Starting over. 40 hours down. (self.stopsmoking)
submitted 7 hours ago by [redacted]
8 comments share

Ahhhhhhhhhhhhhhhhhhh!!!!!!!! (self.stopsmoking)
submitted an hour ago by [redacted] 3 21 days
3 comments share

Fig. 1. Examples of posts made in www.reddit.com/r/stopsmoking.

3 Method

3.1 Data

In this paper Reddit data were chosen since they allow people to create or join communities with a common interest. In particular, data from the largest Reddit community interested in stop smoking – www.reddit.com/r/stopsmoking – were crawled.

The community was founded on 6 November 2009 and as of October 2015, 17,030 users (redditors) have made 33,278 posts in the forum. Figure 1 shows examples of posts made in the community. As seen from the figure, some authors are tagged with their cessation time. There were 8,828 authors who had declared their cessation time. Based on the cessation badge of authors, the cessation days for posts were calculated, which refer to the number of days the authors were abstinent from smoking when making the posts. In this work, we are interested in the posts made from day 1 of current cessation and exclude those made before that day, resulting in 13,566 posts. We categorized these posts into four mutually exclusive subgroups as below to learn how the textual features of the authors change as smoking cessation progresses:

- S0: Within the first week.
- S1: Within the first month (excluding cohort S0).
- S2: From second month to one year.
- S3: Beyond 1 year.

To create a balanced dataset, which is convenient for the evaluation of the classification afterwards, the same number of posts (based on the smallest number of posts, 1,220, which appeared in S3 cohort) for each of the four cohorts was randomly selected into the study, resulting in a corpus of 4,880 posts. This corpus was used in the experiments.

3.2 Feature Sets

In this work, topics, extracted using LDA [1], and language styles, extracted using LIWC package [7], were used to characterize posts made in different stages of quitting smoking. For language styles, LIWC package returns 68 psycholinguistic categories, such as linguistic, social, affective, cognitive, perceptual, biological, relativity, personal concerns and spoken.

Table 1. The accuracy in the two-class classifications of posts into S0 versus later stages by different classifiers with different features. Best results for each feature set are shown in bold.

LIWC	S1	S2	S3	**Topic**	S1	S2	S3	**Joint LIWC & Topic**	S1	S2	S3
Lasso	**54.5**	**62.1**	**71.1**	Lasso	**60.2**	64.3	61.3	Lasso	60	**64.3**	**75.4**
LR	51	59.4	68.6	LR	**60.2**	**64.8**	**63.7**	LR	57.4	61.9	69.9
NB	50.2	59.2	59	NB	59.4	60.9	60.7	NB	**61.9**	63.5	60.7
SVM	49.8	54.7	55.5	SVM	49.2	49.6	49.4	SVM	48.8	54.3	54.9

3.3 Classifiers

Our experimental design examines the effect of topics and language styles in classifying a post into one of four different stages of quitting smoking. We are interested in not only which sets of features perform well in the classification but also which features in the sets are strongly predictive of the cessation stages. For this purpose, the least absolute shrinkage and selection operator (Lasso) [3], a regularized regression model, is chosen. Lasso does logistic regression and selects features simultaneously, enabling an evaluation on both the classification performance and the importance of each feature in the classification. Particularly, in prediction of S0 versus S1, S2, and S3 stages for a post, Lasso assigns positive weights to features more likely to be used in S0 and negative weights to those less to be used in S0. To the features irrelevant to the prediction, Lasso assigns zero weight. Thus, by examining its weights, we can learn the importance of each feature in the prediction.

For comparison with the classification performed by Lasso, classifiers from other paradigms were also included: Naive Bayes (NB), Support vector machines (SVM), and Logistic regression (LR). These classifiers will perform the binary classifications of posts into either S0 or S1/S2/S3 stages, using LIWC, topics, and a combination of them as the feature sets. The accuracy is used to compare with that of Lasso on the same classification.

4 Classification

4.1 Performance

The Lasso model [3] is used for the classification. Using the coefficients derived from the Lasso method, we implemented three pair-wise classifiers classifying

Table 2. Lasso model with language styles as the features to discriminate S0 versus later stages. Features with same coefficient signs across the three binary classifiers were colored. Reds are the positive predictors of S0 and blues are the positive predictors of S1, S2, and S3.

Feature	Example	S1	S2	S3	Feature	Example	S1	S2	S3
(Intercept)		−0.59	−0.45	−0.75	Anger	Hate, kill	0.21		
Word count		0.67	0.31		Sadness	Crying, grief	−0.96		
Words per sentence		−1.92	0.14	6.94	Cognitive	Cause, know	0.14		
Dictionary words		0.95			Insight	Think, know	−0.73		
Words > 6 letters		0.66			Causation	Because, effect	−0.78		
Total function words		−0.64			Discrepancy	Should, would	−0.04		
1st pers singular	I, me	1.83	1.45	1.45	Tentative	Maybe, perhaps	0.38	0.63	
1st pers plural	We, us	−0.24	−0.12		Certainty	Always, never	−1.35	−2.3	
2nd person	You, your	−0.57	−2.11	−3.93	Inhibition	Block, constrain	0.28		
3rd pers singular	She, her	−0.99			Inclusive	And, with	0.55		
3rd pers plural	They, their	−0.24			Exclusive	But, without	−0.31		
Articles	A, an	−0.09			Perceptual	Observing, heard	−1.2		
Auxiliary verbs	Am, will		0.14	0.37	See	Appearance, look	−0.52	−1.26	−0.24
Past tense	Went, ran	−0.61	−0.5		Hear	Listen, hearing	0.63	0.71	
Present tense	Is, does	−0.37			Feel	Feels, touch	1.88	1.12	1.16
Future tense	Will, gonna	1.1			Biological	Eat, blood	−0.91		
Adverbs	Very, really	−1	−0.2		Body	Cheek, hands		0.14	0.82
Prepositions	To, with	0.57	0.13		Health	Clinic, flu	−0.36		
Conjunctions	And, but	0.08	0.57	0.51	Sexual	Horny, love	−0.99		
Negations	No, not	−0.08			Time	End, until	0.53		
Quantifiers	Few, many	−0.06	0.48		Work	Job, majors	1.38	0.5	
Swear words	Damn, piss	3.74	0.24		Achievement	Earn, hero	0.16	0.82	
Social	Mate, talk		−0.07	−0.07	Leisure	Cook, chat	−1.63	−0.06	
Humans	Adult, baby	0.12	−0.06		Home	Apartment, kitchen	0.03		
Positive emotion	Love, nice	−0.56	−0.46		Money	Audit, cash		−1.73	−1.29
Negative emotion	Hurt, ugly	0.46	0.26	1.04	Religion	Altar, church	1.25		
Anxiety	Worried, fearful	−0.61			Assent	Agree, OK	−2.87	−0.53	

input posts into *S0* versus *S1*, *S2*, or *S3* stages, using three feature sets: LIWC, topics, and a combination of them. The accuracy of this classifier in different feature sets is shown in Table 1, accompanied by that of SVM, NB, and LR. Lasso outperformed other classifiers when LIWC and a combination of LIWC and topics were used as the features, and was second to LR when topics were the features. However, Lasso used a smaller number of features than did the best, LR. So, for the sake of brevity, only results by Lasso are reported hereafter.

In general, the result of the classification by Lasso is better when the gap of the stages is wider. In other words, the performance of S0 versus S3 classification is the best, that of S0 versus S2 is in between, and that of S0 versus S1 is the worst. A possible reason is that as cessation progresses the topics and language styles of people in later stages are markedly different from those expressed during the first stage. An exception is the drop in performance on using topics as features to classify posts into S0 versus S3 stages, implying a similarity in the topics discussed by both junior and senior abstainers. It could be because the seniors may talk about their early days of the journey or advise novices on what they may face in the battle, making the use of topics in stage S0 and S3 indistinguishable.

Topics outperformed language styles as the features in S0 versus S1 and S2 classifications. However topics fell behind language styles in the roles of features in S0 versus S3 classification, possibly due to a bigger gap in the use of certain language processes than in the use of topics between the two categories.

Observably, a fusion of the features gained the best performance in S0 versus S3 classification. This shows the potential of using multi-cues for making prediction of people in different stages of quitting an addiction.

4.2 Linguistic Features as the Predictors

Table 2 shows the Lasso model using language style cues as features to predict S0 versus S1, S2, and S3 posts. Obviously, *negative* emotion is a positive predictor of posts made in the first week of smoking abstinence. Likewise, it is also observed that *first personal singular pronouns* is another positive predictor of first-week posts while, *second personal pronouns* is an indicator of posts made in later stages.

An interesting observation is that *feel(ing)* is a positive predictor of S0, while *see(ing)* (e.g., *appearance*, *look, weight*) is a negative predictor of this early stage. This is understandable given that those in the initial stage of quitting tend to experience the feeling of craving, whereas for the seniors they tend to look back and talk about *before and after*, for example, commenting on their appearances, such as *"Skin looks better - Before I'd look pale, dry, wrinkly, like a dying man. Now I feel I have lesser wrinkles. Skin looks more alive. Must also be due to the fact that I'm much better hydrated now..."* or *"You'll only gain weight if you let yourself. If you don't eat junk food as a replacement, you won't gain weight. It's really that simple..."*.

4.3 Topics as the Predictors

Table 3 shows the classification model inferred by Lasso [3] to predict S0 versus S1, S2, and S3 posts using topics as features.[1]

Table 4 shows the word cloud of topics with the same sign of coefficients for all classifications. Strong positive predictors of S0 abstainers include topic numbered 18 about the theme of determination of quitting ("quit", "decided"), topic 23 (the failure in past attempts – "failed", "past", "previous", "attempts"), and topic 47 (methods of quitting smoking – "gum", "vaping"). Other strong predictors of S0 include topic 15, which is on looking for generic advice in Reddit, such as on posts, comments, or edit. This could be because the abstainers in the first week are novices to Reddit.

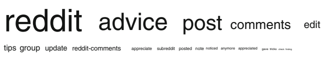
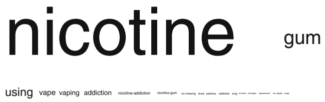
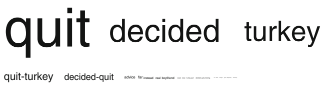
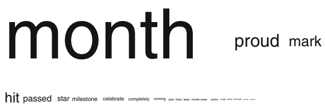
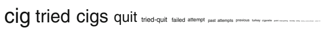
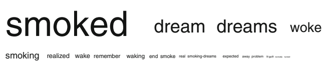
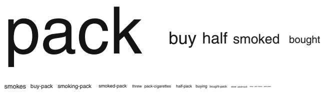
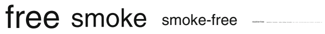
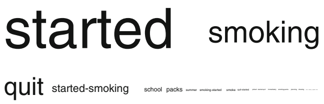
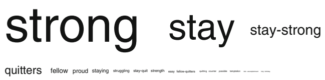
On the other hand, strong positive predictors of posts made by abstainers in later stages are feeling proud after succeeding months of smoking abstinence (topic 10 – "month", "proud", "passed", "hit", "mark"), being happy about winning the challenge (topic 20 – "smoke-free", "cigarette-free", "nicotine-free", "challenge", "glad"), or struggling to deal with the temptation and staying strong (topic 26 – "stay-strong", "struggle", "temptation").

[1] The list of all topics can be accessed via http://bit.ly/21Z0o4r.

Table 3. Lasso model with topics as the features to discriminate posts made in S0 versus those made in later stages of quitting smoking. Features with the same coefficient signs across the three classifiers were colored. Those in red are the positive predictors of S0 and those in blue are the positive predictors of S1, S2, and S3.

Feature	S1	S2	S3	Feature	S1	S2	S3	Feature	S1	S2	S3
(Intercept)	0.36	−0.13	−0.24	T18	0.39	1.38	1.14	T35		0.14	
T1			−1.34	T19	−0.14	−0.66	−0.51	T36		−0.05	−0.53
T2		0.01	−0.27	T20	−0.04	−0.74	−0.95	T37	1.26	−2.12	−0.47
T3			−1.15	T21		−0.89	−0.94	T38		0.37	2.01
T5		1.37	0.36	T22		−0.05		T39	−0.41		−0.44
T6			0.21	T23	0.37	2.49	2.72	T40			−0.85
T8			−0.79	T24	0.75	0.82	0.75	T41			0.38
T9		−0.33	−0.44	T25	−3.55	−1.03		T42			−1.59
T10	−0.09	−2.26	−1.19	T26	−0.52	−0.85	−2.17	T43		0.74	1.14
T11		0.46		T28		0.71	3.31	T44	−0.43		0.13
T12		−0.11	−1.84	T29		0.35		T45		−0.1	
T13		0.44	0.41	T30	0.16	1.25	0.43	T46			−0.69
T14		0.8	2.83	T31		0.5	1.3	T47	0.48	1.18	1.53
T15	0.06	0.75	0.76	T32		−1.13	−2.26	T48	−0.37		−1.1
T16		0.4	1.57	T33		−0.45	−1.16	T49		−0.43	−0.06
T17			0.99	T34		0.16	1.01	T50			2.8

Table 4. Topics selected into the prediction models with the same coefficient signs in the three classifiers. Red indicate positive predictors of S0 and blue indicates its negative predictors.

Topic	Word cloud	Topic	Word cloud
T15	 reddit advice post comments edit tips group update reddit-comments appreciate subreddit posted note noticed anymore appreciated	T47	 nicotine gum using vape vaping addiction nicotine-addiction nicotine-gum
T18	 quit decided turkey quit-turkey decided-quit advice	T10	 month proud mark hit passed star milestone celebrate completely coming
T23	 cig tried cigs quit tried-quit failed attempt past attempts previously	T19	 smoked dream dreams woke smoking realized wake remember waking end smoke real smoking-dreams expected away problem
T24	 pack buy half smoked bought smokes buy-pack smoking-pack smoked-pack threw pack-cigarettes half-pack buying bought-pack	T20	
T30	 started smoking quit started-smoking school packs summer smoking-started smoke	T26	 strong stay stay-strong quitters fellow proud staying struggling stay-quit strength easy fellow-quitters

5 Conclusion

This study investigated the topics and linguistic features of the discussions among members in an online community interested in quitting smoking. Machine learning techniques were used to discriminate the textual features among posts made in early and late stages of quitting smoking. It was found that distinct topics and linguistic styles differentiate abstainers in the first week from those in later stages, probably providing textual factors for both failure and success in smoking cessation. The results of this study suggest that data mining of low-cost social media has the potential to detect meaningful patterns of addiction problems confronted by society, likely offering an effective tool for policy makers. The findings also highlight the potential applicability of machine learning to health care practice and research.

This work has explored the markers of different stages in the current cessation attempt. Since Reddit does not record the cessation badge for previous attempts, manual annotations could be conducted to collect all the attempts. When this information is available, future work could investigate into the causes of relapsing among abstainers.

Acknowledgment. This work is partially supported by the Telstra-Deakin Centre of Excellence in Big Data and Machine Learning.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
2. Chen, A.T., Zhu, S.-H., Conway, M.: What online communities can tell us about electronic cigarettes and hookah use: a study using text mining and visualization techniques. J. Med. Internet Res. **17**(9), e220 (2015)
3. Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. J. Stat. Softw. **33**(1), 1 (2010)
4. Kumar, M., Dredze, M., Coppersmith, G., De Choudhury, M.: Detecting changes in suicide content manifested in social media following celebrity suicides. In: Proceedings of the ACM Conference on Hypertext & Social Media, pp. 85–94 (2015)
5. Nguyen, T., Phung, D., Adams, B., Venkatesh, S.: A sentiment-aware approach to community formation in social media. In: Proceedings of the International AAAI Conference on Weblogs and Social Media, pp. 527–530 (2012)
6. Nguyen, T., Phung, D., Venkatesh, S.: Analysis of psycholinguistic processes and topics in online autism communities. In: Proceedings of the IEEE International Conference on Multimedia and Expo, pp. 1–6 (2013)
7. Pennebaker, J.W., Francis, M.E., Booth, R.J.: Linguistic Inquiry and Word Count (LIWC) [Computer software]. LIWC Inc. (2007)
8. Tamersoy, A., De Choudhury, M., Chau, D.H.: Characterizing smoking and drinking abstinence from social media. In: Proceedings of the ACM Conference on Hypertext & Social Media, pp. 139–148 (2015)

Query Processing

POL: A Pattern Oriented Load-Shedding for Semantic Data Stream Processing

Fethi Belghaouti[1], Amel Bouzeghoub[1](✉), Zakia Kazi-Aoul[2], and Raja Chiky[2]

[1] SAMOVAR, Telecom SudParis, CNRS, Universite Paris-Saclay,
9 rue Charles Fourier, 91011 Evry Cedex, France
{fethi.belghaouti,amel.bouzeghoub}@telecom-sudparis.eu
[2] Institut Superieur d'Electronique de Paris,
28 rue Notre-Dame des Champs, 75006 Paris, France
{zakia.kazi-aoul,raja.chiky}@isep.fr
http://www.telecom-sudparis.eu, www.isep.fr

Abstract. Nowadays, high volumes of data are generated and published at a very high velocity, producing heterogeneous data streams. This has led researchers to propose new systems named RDF Stream Processors (RSP), to deal with this new kind of streams. Unfortunately, these systems are fallible when their maximum supported speed is reached especially in a limited system resources environment. To overcome these problems, recent efforts have been made in the field. Some of them decrease the volume of RDF data streams using compression or load-shedding techniques, mostly according to a probabilistic approach. In this paper we propose POL: a Pattern Oriented approach to Load-shed data from RDF streams based on a deterministic approach. As a pre-processing task through a unique pass, the approach extracts the exact needed semantic data from the stream. The conducted experiments on public available datasets have demonstrated the effectiveness of our approach.

Keywords: BigData · Semantic data stream · Graph patterns detection · Load-shedding

1 Introduction

In the very near past, queries were volatile when data was persistent. Today, we are witnessing the inversion of roles. Thus, data is becoming extremely dynamic when the queries are persistent. Indeed, data is generated continuously as a stream by different sources such as sensors, social networks, GPS, e-commerce and weather stations to cite only few and is heterogeneous (various formats such as JSON, XML, RSS, CSV, etc.). This fact leads to an interoperability problem.

Today, in order to provide useful information, such as contextual data, for target applications and increase interoperability, initiatives such as the Semantic Sensor Web (SSW)[1] have semanticized their descriptions and their observation data using semantic web technologies [6][2], giving rise to semantic data streams.

[1] http://en.wikipedia.org/wiki/Semantic_Sensor_Web.

[2] http://www.w3.org/standards/semanticweb/.

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 157–171, 2016.
DOI: 10.1007/978-3-319-48743-4_13

However, given the specificity of this type of streams, neither Data Stream Management Systems (DSMS) [1,3] nor standard semantic web technologies were adapted to process this new type of data flows. This has favored the emergence of a new research axis from the semantic web community and led researchers to propose RDF Stream Processing systems (*RSP*) as a solution to deal with this new kind of streams. We cite C-SPARQL [5], CQELS [15], SPARQL Stream [8], Sparkwave [12], EP-SPARQL [2] and Streaming SPARQL [7]. This community has created recently the W3C RSP Group[3] in order to define "a common model for producing, transmitting and continuously querying RDF Streams" (see footnote 3).

Dealing with huge volumes of dynamic data could overload the system, which causes a significant increase in response time and an inevitable degradation of response quality and sometimes even to its unavailability. In addition, none of the proposed RSPs includes yet a quality of service policy (QoS) [13], which makes them fallible when their maximum supported speed is reached or the resources of the system hosting them are saturated. To overcome such situations, in contrast to the existing probabilistic solutions, inspired by the DSMS domain techniques ([17,21,27]), we propose a deterministic approach named POL: a Pattern Oriented approach for Load-shedding semantic data streams. Our approach proposes to use basic boolean operations as a "low cost" pre-processing. After constructing the binary pattern of the continuous query, for each received RDF graph, a conjunction operator (AND) is applied between its online binary detected pattern and the binary representation of the query. This determines, very quickly, the relevant part of the data to send to the RSP engine and the irrelevant ones to load-shed according to the particular query. To implement our solution, we consider the existing RSP systems as black boxes. The objective of our solution is to reduce the volume of the input stream and to optimize the memory and CPU usage increasing thus the system processing capacity and ensuring its availability while guaranteeing its recall at 100 %.

The remainder of the paper is organized as follows: Sect. 2 presents the related work, giving a critical overview of existing approaches for decreasing the load on RDF Stream Processors. We present the Pattern Oriented Load-shedding approach in Sect. 3. Section 4 reports the empirical evaluation. Finally, we conclude and give some research perspectives in Sect. 5.

2 Related Work

Obviously, processing and storing the entire data of a stream, which is an infinite set of tuples is impossible, in particular, if the system has a restricted set of processing and storage resources. To deal with those constraints, two types of approaches have been adopted by researchers.

The first type, which has a financial impact, considers the elasticity of the system, using Cloud Computing technologies to allocate as many resources as necessary. As far as we know, there are two works dealing with elasticity.

[3] http://www.w3.org/community/rsp/.

In [18], Hoeksema and Kotoulas implement partial RDFS reasoning as part of the C-SPARQL query language on the S4 streaming platform[4], which enables to split the processing load over multiple machines to increase the overall system throughput. In [19], Le Phuoc et al. propose CQELS Cloud which allows nodes to join or leave, and re-assigns operators to nodes accordingly.

The second type of existing approaches has no financial impact and considers the inflexible and rigid aspects of the system, where the only resources to allocate are the ones that are physically available. Consequently, when the data stream volume and/or rate reach some maximum threshold, the system could be saturated causing a significant degradation in response's quality and time and probably makes the system itself unavailable. To overcome such situations, this type of approaches uses techniques from DSMS domain [4,10,17,21] which consist of reducing the input load by shedding a part of its data to avoid the system resources saturation.

In their work, Jain et al. in [20] have extended CQELS by adding new operators and implementing three sampling algorithms: Uniform Random Sampling, Reservoir Sampling and Chain Sampling. Depending on the algorithm and the sampling rate passed as parameters to the query, this approach consists in ignoring the RDF triple or passing it to the target system.

In [14], Nguyen et al. propose eviction strategies for semantic flow processing by dropping variable bindings instead of data, unlike what is done in load shedding approaches. This probabilistic eviction is based on the fact that a query is represented by a tree of algebra expressions. The variable bindings are propagated from the leaves to the root, representing the result. Each algebra expression has a cache where the variable bindings are stored. Thus, based on the probability that its result set is not empty, the algorithm decides to evict a binding. Nevertheless, these probabilities are computed offline, which means that they must be calculated beforehand. Moreover, this strategy does not take into account the structure of the graph.

In [11], authors propose CLOCK, a data-aware eviction strategy that extends Last Recently Used (LRU) algorithm. Indeed, it considers not only the last time the variable binding has been used, but also the importance of past usefulness in order to estimate the likelihood of a future one and evict bindings from the cache based on this estimation. In this approach, every binding is associated with a score. The scores are stored in a circular buffer, and a pointer points at the position p. If the cache is full, then the score at position p is depreciated and if it is lower than a chosen threshold, the corresponding element is evicted. If not, the pointer moves to the next element and so on until an element gets evicted. If a binding contributes to a join, its score is increased. This strategy is more efficient than LRU, but it needs an extra-buffer.

These techniques are recently used in RSP community (see footnote 3). Even if they avoid systems overload and/or crash, they however decrease the quality of their responses trying to maximize their recall. In addition, they are all based on a probabilistic data-oriented approach and some of them need off-line computations.

[4] http://incubator.apache.org/s4/.

To overcome these disadvantages, we proposed in a previous work [24], a graph-oriented approach for load-shedding semantic data streams. The main idea of this work is to prove that semantic data streams should be processed as a stream of sub-graphs instead of triples. We have shown that the graph-oriented approach preserves the links between data which leads to a higher semantic level and best performances comparing to the triple-based approach which destroys the links between nodes during the load-shedding and thus decreases the semantic level of the RDF stream. Table 1 summarizes and compares the existing works in the literature according to some criteria such as the handled data vs graph structure, processing type (online vs offline) and whether the method is probabilistic or deterministic.

Table 1. Comparative study of existing works.

Approaches	Data vs structure	Offline vs online	Probabilistic vs deterministic
Jain et al. [20]	Data	Online	Probabilistic
Gao et al. [11]	Data	Online	Probabilistic
Nguyen et al. [14]	Data	Offline	Probabilistic
Belghaouti et al. [24]	Graph	Online	Probabilistic
POL	Graph	Online	Deterministic

In this paper, we propose POL: a Pattern Oriented approach for Load-shedding semantic data streams. This new approach, that guarantees a 100 % of the system recall, is based on an exact matching between the continuous query pattern and the input stream one using Boolean operations. We detail this algorithm in the next section.

3 Pattern Oriented Load-Shedding: POL

Our approach performs the load-shedding through three steps:

- (1) The online predicates pattern detection and their hash table (PHT) construction detailed in Sect. 3.2,
- (2) The RDF graph and continuous query bit vectors construction presented in Sect. 3.3, and
- (3) The load-shedding mechanism described in Sect. 3.4.

Before detailing these steps, we first give some necessary definitions that will be used in the following subsections.

3.1 Definitions

Let $\mathcal{S} = \{G_1, .., G_n\}$ be an RDF stream where each graph G_i is a finite set of RDF triples:

$$G_i = \{(s_{i1}, p_{i1}, o_{i1}), (s_{i2}, p_{i2}, o_{i2}), ..., (s_{im}, p_{im}, o_{im})\}$$

We assume that every complex tree-based RDF graph can be divided into a set of star graphs with a unique subject [25].

The graph is thus reduced to a set of triples having the same subject:

$$G_i = \{(s_i, p_{i1}, o_{i1}), (s_i, p_{i2}, o_{i2}), ..., (s_i, p_{im}, o_{im})\}$$

Hence, we can formalize our understanding of a graph in an RDF stream and the notions of graph and query patterns as follows:

Property 1. Each graph in an RDF Stream can be represented as a directed star-graph $G_i(V, E)$, where $V = \{v_0, v_1, v_2, ..., v_m\}$ is the set of vertices with v_0 the central vertex and vi the leaf vertices for $i = 1..m$ and $E = \{(v_0, v_1), (v_0, v_2), ..., (v_0, v_m)\}$ is the set of edges labeled with the predicates.

Definition 1 ***(Graph Pattern).*** *Let $P = \{p_1, ..., p_n\}$ be the set of predicates in the graph stream $\mathcal{S}$, $GP_i = \{p_k \in P \mid k \leq n\}$ a subset of P and $G_i \in \mathcal{S}$ a graph in the stream.*

$$GP_i \textit{ is a graph pattern of } G_i \; iff$$

$$\forall p_k \; (p_k \in GP_i \rightarrow p_k \in G_i)$$

Definition 2 ***(Query Pattern).*** *Let $P = \{p_1, ..., p_n\}$ be the set of predicates in the graph stream $\mathcal{S}$, $QP_j = \{p_l \in P \mid l \leq n\}$ a subset of P and Q_j a continuous query.*

$$QP_j \textit{ is a query pattern of } Q_j \; iff$$

$$\forall p_l \; (p_l \in QP_j \rightarrow p_l \in Q_j)$$

3.2 Frequent Predicates Detection and PHT Construction

The pseudo Algorithm 1 explains how we construct the Predicate Hash Table (*PHT*) by analyzing the predicates of the RDF stream. This table contains all the detected predicates in RDF graphs of an input stream. *PHT* is an indexed table where each new detected predicate is inserted. When the PHT is empty at the beginning of the stream (Algorithm 1, line 2), each time an RDF graph is received (Algorithm 1, line 4), for all its predicates, if the predicate is not present in the *PHT*, it is inserted with a new index (Algorithm 1, lines 5 to 14). Note that those indexes will serve later to point the bits in the *GraphBV* and the *QueryBV*.

Algorithm 1. Frequent RDF Predicates Detection

```
Data: RDF Stream
Result: Predicates Hash Table (PHT)
begin
    HashTable PHT<predicate, index>
    int i ← 0                                      /* Index initialization */
    foreach $graph \in RDF\ Stream$ do
    foreach $predicate \in graph$ do
    begin
        if $predicate \in PHT$ then
            NOP                                 /* already existing predicate */
        else
            ind ← i
            PHT.put(predicate, ind)              /* Insert the new predicate */
            i ← i+1                                 /* Update the bit index */
        end
    end


    return PHT
end
```

3.3 Query and Graph Bit Vectors Construction

As stated in Sect. 3.1, since each RDF graph can be considered as a set of RDF star graphs, each graph can be represented as a bit vector that we call *GraphBV*. Each bit at index i of this vector is set to 1 or 0 according to the presence or not of the corresponding predicate in the graph. Figure 1 illustrates how the two bit vectors are constructed. In this example, the graph contains the predicates a, b, c, d; and the query contains the predicates a and c.

We consider in this section two bit vectors: *GraphBV* with size m associated to each graph pattern GP and *QueryBV* with size k associated to each graph query GQ. p_i and p_j are predicates belonging to PHT.

$$\forall\ i \in [0, m-1],\ \ GraphBV[i]\ \ = \begin{cases} 1 \text{ if } p_{i+1} \in GP \\ 0 \text{ else.} \end{cases}$$

And

$$\forall\ j \in [0, k-1],\ \ QueryBV[j]\ \ = \begin{cases} 1 \text{ if } p_{j+1} \in \text{GQ} \\ 0 \text{ else.} \end{cases}$$

When a query is received by the RSP, the *QueryBV* is initially set to zero (Algorithm 2, line 2). Then, for each predicate of the query, the corresponding bit is set to 1 (according to the PHT). If the predicate does not exist in the PHT, it is inserted as new predicate pattern and its corresponding bit is set to "1" (Algorithm 2, lines 3 to 12). The bit vector *GraphBV*, associated to the RDF

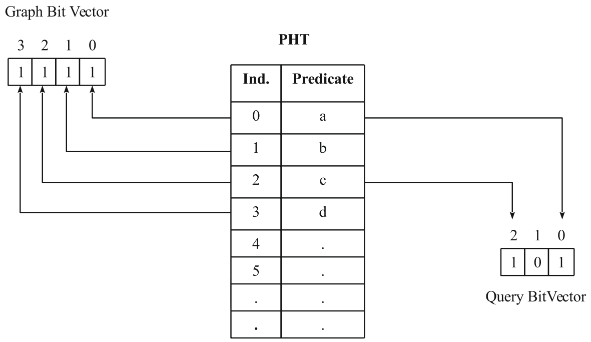


Fig. 1. Construction of the query and graph BitVectors (patterns).

Algorithm 2. Query Pattern Detection

```
Data: RSP Continuous Query, Predicate Hash Table (PHT)
Result: Query Bit Vector (QueryBV)
begin
    BitVector QueryBV ← 0                              /* Query bitvector initialization */
    foreach predicate ∈ Query do
    begin
        if predicate ∈ PHT then
            ind ← PHT.get(predicate)                    /* Get the predicates bit index */
            QueryBV[ind] ← 1                            /* Set the relevant bit to 1 */
        else
            Insert(predicate) into PHT                  /* Insert the new predicate */
            Update(ind)
            QueryBV[ind] ← 1                            /* and set the relevant bit to 1 */
        end
    end

    return QueryBV                                      /* Return the Query Pattern */
end
```

graph received from the stream, is constructed in the same way (the algorithm is not presented to avoid redundancy).

3.4 The Load-Shedding

Our Pattern oriented Load-Shedding approach consists in a continuous process that filters the incoming RDF stream, thus keeping only the requested ones for the RSP engine. Based on using bitwise operations like the conjunction *(binary AND)* and hash table accesses, our approach avoids the RSP system doing complex operations on RDF graphs that are usually very "costly".

Pseudo Algorithm 3 explains how, for each graph in the RDF stream (line 3), we drop all the triples that contain predicates which are not requested by the query. The only ones that are transmitted (not dropped) to the RSP engine are those having their index-bits set to 1 in both *GraphBV* and *QueryBV* (lines 7,10).

Algorithm 3. Pattern Oriented Load-Shedding

```
Data: RDF Stream, QueryBV, PHT
Result: Load-Shedded RDF Stream
begin
    Query Pattern Detection(QueryBV)
    foreach graph ∈ RDF Stream do
    begin
        foreach triple t ∈ graph do
        begin
            if t.predicate ∈ PHT then
                ind ← PHT.get(t.predicate) /* Get the existing predicates
                bit index */
                if QueryBV[ind] = 1 then
                    NOP                                /* Keep the triple */
                else
                    drop(t)   /* The irrelevant triple is Load-Shedded */
                end
            else
                drop(t)       /* The irrelevant triple is Load-Shedded */
            end
        end

        return graph             /* return the lightweighted graph  */
    end

end
```

3.5 Proof of Concept: Load-Shedding Semantic Data Streams Using the Graph (data) and Query Patterns

Let suppose that we have a continuous query *q* as in listing 1.1.

```
SELECT ?x ?y ?z
FROM STREAM <http://MyStream> [NOW 10s SLIDE 10s]
WHERE {?x a ?y ;
          c ?z .
      }
```

Listing 1.1. Continuous query example

Figure 2 illustrates the content of the current window that contains the data received from the stream during the last ten seconds, as mentioned in the query *q* (Listing 1.1). Our approach consists of constructing the patterns of the query and its bit vector (once). Then, every time an RDF graph is received, the Load-Shedding algorithm computes a Boolean operation AND between those two bit vectors *GraphBV* and *QueryBV* dealing to update the *GraphBV* value ($GraphBV'$ in Fig. 2). According to the result, we keep only triples of the

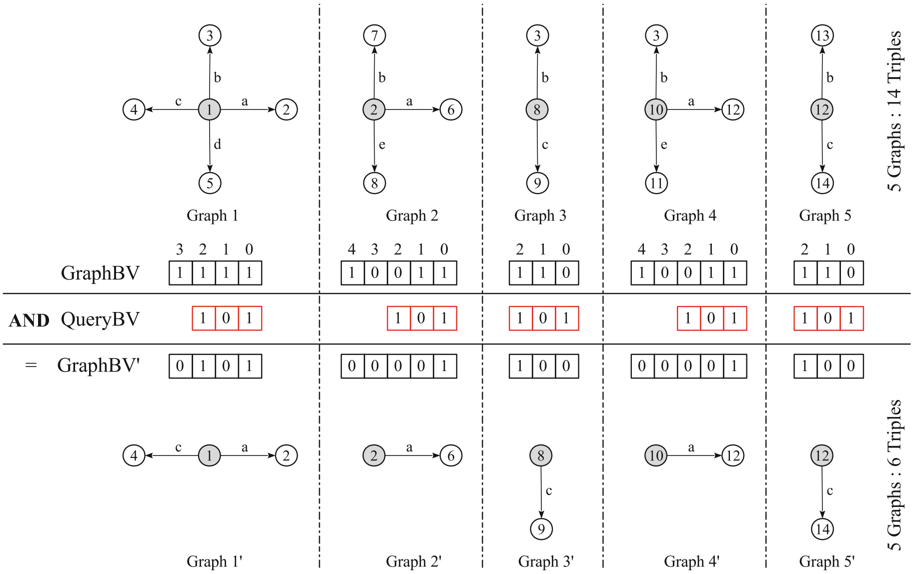


Fig. 2. RDF data stream example: applying pattern oriented load-shedding.

incoming RDF graph that contain the predicates of the query. This operation reduces the memory consumption, enhances the system workload and of course guarantees its good quality of response.

Thus, Fig. 2 shows that with a $QueryBV$=101, the algorithm reduces the volume of data in the current window from 5 graphs containing 14 triples to 5 graphs containing only 6 triples. For example, Graph1 is reduced from $\{(1, a, 2), (1, b, 3), (1, c, 4), (1, d, 5)\}$ to $\{(1, a, 2), (1, c, 4)\}$ and graph2 is reduced from $\{(2, a, 6), (2, b, 7), (2, e, 8)\}$ to $\{(2, a, 6)\}$ and so on.

4 Evaluation and Discussion

POL algorithm has been implemented using Java language on the top of Ubuntu-64 14.04 LTS OS and a personal laptop (Intel core i7-4500 4X1,8 GHz with 6 GB of RAM). For the experimentation purpose, we choose C-SPARQL as an RSP engine. However, as explained previously, any other RSP engine could be used. In order to assess the effectiveness of our approach, we launch C-SPARQL queries twice (1st time with POL and the 2nd time without POL).

In this section we present the Key Performance Features that we aim to achieve, the 5 public datasets used in the conducted experimentations, and a case study to illustrate the proposal on a real extracted semantic data stream. Finally we will explain the evaluation results and give a final discussion.

4.1 Performance Key Features

Before presenting the conducted experimentation, we detail here the three main Performance Key features that our approach must improve. These keys are useful to highlight the contribution of our approach that is to say, enhancing the efficiency of an RSP engine in a restricted system resources environment. We describe these performance keys in the following:

- Time Efficiency Feature (TE). A system is more efficient in time than another, if it can do the same task in less time. Our approach can enhance the efficiency of an RSP engine by reducing this key feature when applying a single pass load-shedding operation to avoid processing irrelevant data by the engine, improving thus its time efficiency and the system scalability. This metric is computed as follows:

$$Time\ Efficiency\ in\%\ (TE) = \frac{ET}{ETLS}$$

  where ET and $ETLS$ are the execution time of the engine without and with POL respectively.
- Space Efficiency Feature (SE). A system is more efficient in space than another, if it can do the same task using less memory space. Our approach decreases the memory space system usage by load-shedding an extract of input data stream (irrelevant data). It enhances the space efficiency, and thus the system scalability. This feature is the ratio between the number of the triples processed by the engine without and with Load-Shedding (POL). It is computed as follows:

$$Space\ Efficiency\ in\%\ (SE) = \frac{NTriples}{NTriples - NLSTriples}$$

  where $NTriples$ and $NLSTriples$ are respectively the total Number of Triples in the stream and the Number of Load-Shedded Triples using POL.
- Recall Feature (SRecall). The system Recall is usually defined as the number of correct returned results divided by the number of all the relevant results. When applying a probabilistic load-shedding algorithm to a data stream, the recall of the target system is necessarily reduced. Our approach can ensure a 100 % recall, even if -depending of the query- more than half of the volume of the input data is shedded. We define SRecall (Stream Recall) as a specific recall where its value is the result of the division of the number of the tuples returned by the engine when using POL by the one without using it. This key shows how POL can preserve the quality of response of the system.

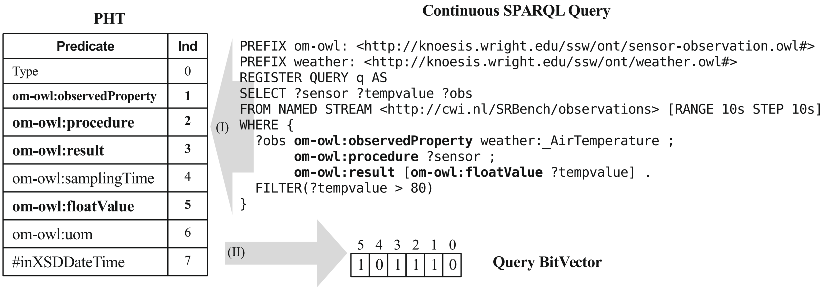
$$SRecall = \frac{NRLS}{NR} \times 100\,\%$$

4.2 Datasets Description

AEMET-1 and AEMET-2 are two datasets provided by the Spanish Meteorological Office (AEMET). They represent meteorological information, taken from weather stations in Spain [9] according to different schemas. The Petrol dataset provides metadata about credit cards transactions in petrol station, furnished by a Spanish start-up (Localidata[5]). Charley and Katrina are two datasets within others delivered by Linked Observation Data (LOD)[6]. They represent sensor observations of different weather parameters. Those observations represent meteorological phenomena like humidity, temperature, pressure, visibility, precipitation, etc.

4.3 Case Study

Figure 3 presents an example of a C-sparql query and illustrates the resulted Query Bit Vector (*QueryBV*) based on the PHT. As explained in Sect. 3.3, Algorithm 2 constructs it by checking the presence of the query predicates one by one (I) and setting to 1 each corresponding bit (here indexes 1, 2, 3 and 5). The returned value is thus 101110 (46) (II).



PHT

Predicate	Ind
Type	0
om-owl:observedProperty	**1**
om-owl:procedure	**2**
om-owl:result	**3**
om-owl:samplingTime	4
om-owl:floatValue	**5**
om-owl:uom	6
#inXSDDateTime	7

Continuous SPARQL Query

```
PREFIX om-owl: <http://knoesis.wright.edu/ssw/ont/sensor-observation.owl#>
PREFIX weather: <http://knoesis.wright.edu/ssw/ont/weather.owl#>
REGISTER QUERY q AS
SELECT ?sensor ?tempvalue ?obs
FROM NAMED STREAM <http://cwi.nl/SRBench/observations> [RANGE 10s STEP 10s]
WHERE {
  ?obs om-owl:observedProperty weather:_AirTemperature ;
       om-owl:procedure ?sensor ;
       om-owl:result [om-owl:floatValue ?tempvalue] .
  FILTER(?tempvalue > 80)
}
```

5	4	3	2	1	0
1	0	1	1	1	0

Fig. 3. Example of query bit vector construction using the PHT table.

Figure 4 shows an extract of an RDF stream on which we apply our approach. At the right side the evolution of the RDF stream over the time is depicted. At the left side, we can see the constructed *GraphBV* corresponding to each graph pattern and the new graph bit vector corresponding to the results of the *AND* operation between *GraphBV* and *QueryBV* for each graph of the stream (*GraphBV AND QueryBV*). All data in red will be dropped. The RSP engine will receive only the relevant data (in black), which will contribute significantly to save space and time.

4.4 Evaluation Results

Table 2 lists the experimental datasets, reporting: number of triples (# RDF Triples), number of RDF graphs (# RDF Graphs), the space efficiency key (SE), the time efficiency key feature (TE) and finally the SRecall.

[5] http://www.localidata.com/.

[6] http://wiki.knoesis.org/index.php/LinkedSensorData#Linked_Observation_Data.

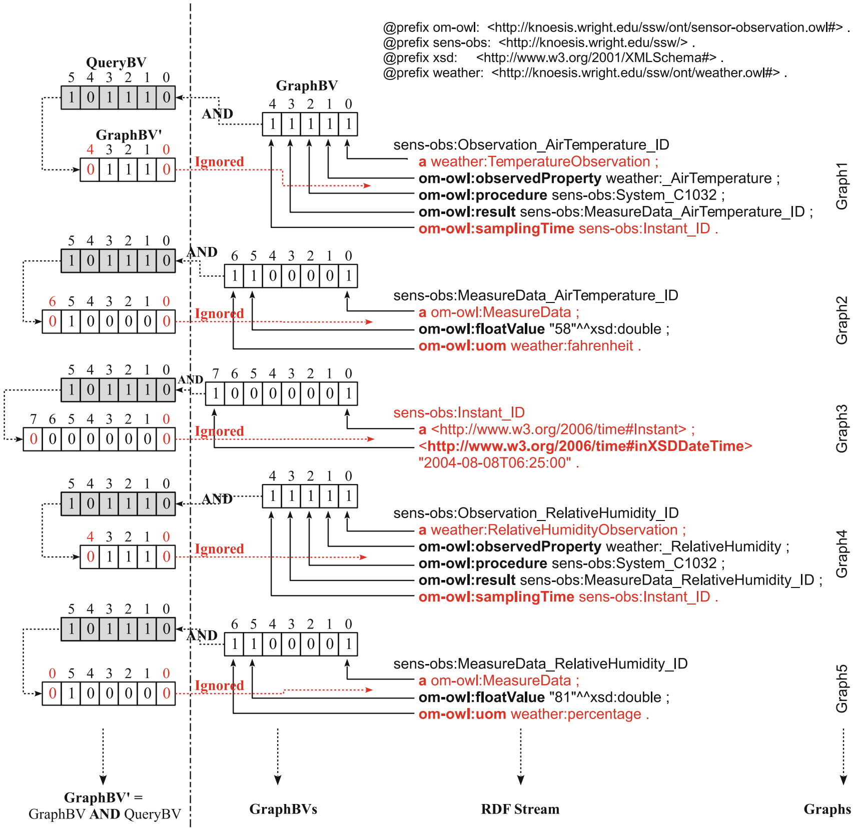

@prefix om-owl: <http://knoesis.wright.edu/ssw/ont/sensor-observation.owl#> .
@prefix sens-obs: <http://knoesis.wright.edu/ssw/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix weather: <http://knoesis.wright.edu/ssw/ont/weather.owl#> .
QueryBV
GraphBV
GraphBV'
AND
Ignored
sens-obs:Observation_AirTemperature_ID
a weather:TemperatureObservation ;
om-owl:observedProperty weather:_AirTemperature ;
om-owl:procedure sens-obs:System_C1032 ;
om-owl:result sens-obs:MeasureData_AirTemperature_ID ;
om-owl:samplingTime sens-obs:Instant_ID .
Graph1
sens-obs:MeasureData_AirTemperature_ID
a om-owl:MeasureData ;
om-owl:floatValue "58"^^xsd:double ;
om-owl:uom weather:fahrenheit .
Graph2
sens-obs:Instant_ID
a <http://www.w3.org/2006/time#Instant> ;
<http://www.w3.org/2006/time#inXSDDateTime>
"2004-08-08T06:25:00" .
Graph3
sens-obs:Observation_RelativeHumidity_ID
a weather:RelativeHumidityObservation ;
om-owl:observedProperty weather:_RelativeHumidity ;
om-owl:procedure sens-obs:System_C1032 ;
om-owl:result sens-obs:MeasureData_RelativeHumidity_ID ;
om-owl:samplingTime sens-obs:Instant_ID .
Graph4
sens-obs:MeasureData_RelativeHumidity_ID
a om-owl:MeasureData ;
om-owl:floatValue "81"^^xsd:double ;
om-owl:uom weather:percentage .
Graph5
GraphBV' =
GraphBV AND QueryBV
GraphBVs
RDF Stream
Graphs

Fig. 4. Pattern oriented load-shedding on LOD data stream extract.(Black: kept data, Red: Shedded data). (Color figure online)

We can clearly notice how POL contributes on scaling up an RSP by decreasing its memory consumption more than 11 times (aemet-1 dataset for example) and similarly multiplying its execution speed by approximatively the same rate.

Table 2. Pattern oriented load-shedding approach applied on different datasets.

DataSet	# RDF triples	# RDF graphs	SE	TE	SRecall
aemet-1	1 018 815	33 095	**11,10**	**11,73**	**101 %**
aemet-2	2 788 429	398 347	**3,50**	**3,50**	**101 %**
Petrol	3 356 616	419 577	**4,00**	**4,03**	**98 %**
Charley	108 644 569	25 303 346	**2,34**	**2,32**	**110 %**
Katrina	179 128 408 (*944 510)	41 600 926	**1,30**	**1,31**	**103 %**

Depending on the size of RDF graphs in the stream and the graph of the continous query, the ratio of the data to ignore could be different. The load-shedding is inversely proportional to the query graph size and directely proportional to the RDF graphs size. For example in the aemet-1 dataset, we obtain the higher value of Space and Time efficiencies since the query asks only three predicates while RDF graphs may contain 59 predicates. This explains the high value of the ratio of ignored data. In contrast, in the Charley dataset, the size of RDF graphs is most time less than five predicates, while the query graph size is four predicates. In this case most of the data are passed to the RSP engine.

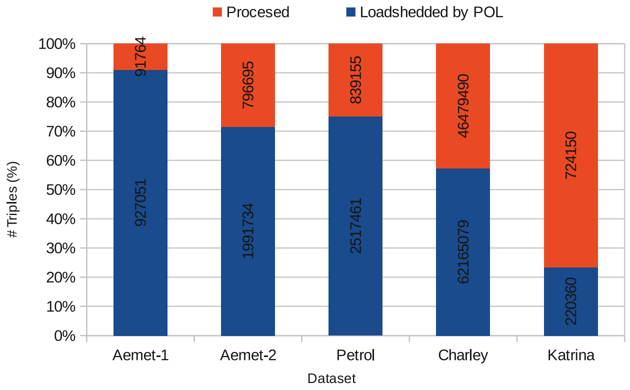


Fig. 5. Space efficiency results.

Figure 5 highlights space efficiency results and illustrates the processed triples vs. the unprocessed ones (dropped by POL). As we can see, the system supported by POL may process only relevant data (in red). This result directly affects in a positive way time efficiency of the system. The other interesting result is that the SRecall is between 98 % and 110 %. This is explained by the fact that (i) the stream processor may give different valid results (unlike a DBMS) and (ii) our approach for load-shedding reduces the probability to lose data during the windowing process.

5 Conclusion

We present in this paper a new deterministic approach for load-shedding RDF data streams named POL. Our Pattern Oriented Load-shedding approach is able to increase the scalability of any RSP system at least 1.3 times to more than 11 times the space and time efficiency without any degradation on the quality of responses. Moreover, it increases the number of the engine answers, thus, its quality of services. All those contributions are based on very "low cost" operations as a boolean operation and a hash table access. In our future works, we will explore the scalability of RSPs in an unlimited system resources environment. We plan to use Cloud Computing technologies to offer them horizontal scalability.

Acknowledgments. This work is partially funded by the French National Research Agency (ANR) project CAIR (ANR-14-CE23-0006).

References

1. Abadi, D., Carney, D., Cetintemel, U., Cherniack, M., Convey, C., Erwin, C., Galvez, E., Hatoun, M., Maskey, A., Rasin, et al.: Aurora: a data stream management system. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (2003)
2. Anicic, D., Fodor, P., Rudolph, S., Stojanovic, N.: EP-SPARQL: a unified language for event processing and stream reasoning. In: Proceedings of the 20th International Conference on World Wide Web, WWW 2011, pp. 635–644. ACM, New York (2011)
3. Arasu, A., Babcock, B., Babu, S., Datar, M., Ito, K., Nishizawa, I., Rosenstein, J., Widom, J.: Stream: the stanford stream data manager (demonstration description). In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 665–665. ACM (2003)
4. Babcock, B., Datar, M., Motwani, R.: Load shedding for aggregation queries over data streams. In: 2004 Proceedings of 20th International Conference on Data Engineering, pp. 350–361, March 2004
5. Barbieri, D.F., Braga, D., Ceri, S., Grossniklaus, M.: An execution environment for c-SPARQL queries. In: Proceedings of the 13th International Conference on Extending Database Technology, EDBT 2010, pp. 441–452. ACM, New York (2010)
6. Berners-Lee, T., Hendler, J., Lassila, O., et al.: The semantic web. Sci. Am. **284**(5), 28–37 (2001)
7. Bolles, A., Grawunder, M., Jacobi, J.: Streaming SPARQL - extending SPARQL to process data streams. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 448–462. Springer, Heidelberg (2008). doi:10.1007/978-3-540-68234-9_34
8. Calbimonte, J.-P., Corcho, O., Gray, A.J.G.: Enabling ontology-based access to streaming data sources. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 96–111. Springer, Heidelberg (2010)
9. Corcho, Ó., Garijo Verdejo, D., Mora, J., Poveda Villalon, M., Vila Suero, D., Villazón-Terrazas, B., Rozas, P., Atemezing, G.A.: Transforming meteorological data into linked data. Semantic Web (2012)
10. Das, A., Gehrke, J., Riedewald, M.: Approximate join processing over data streams. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 40–51. ACM (2003)
11. Gao, S., Scharrenbach, T., Bernstein, A.: The clock data-aware eviction approach: towards processing linked data streams with limited resources. In: Presutti, V., d'Amato, C., Gandon, F., d'Aquin, M., Staab, S., Tordai, A. (eds.) ESWC 2014. LNCS, vol. 8465, pp. 6–20. Springer, Heidelberg (2014)
12. Komazec, S., Cerri, D., Fensel, D.: Sparkwave: continuous schema-enhanced pattern matching over RDF data streams. In: DEBS, pp. 58–68. ACM (2012)

13. Margara, A., Urbani, J., van Harmelen, F., Bal, H.: Streaming the web: reasoning over dynamic data. Web Semant.: Sci. Serv. Agents World Wide Web **25**, 24–44 (2014)
14. Nguyen, M.K., Scharrenbach, T., Bernstein, A.: Eviction strategies for semantic flow processing. In: SSWS@ ISWC, pp. 66–80 (2013)
15. Phuoc, D.L.: A native and adaptive approach for linked stream data processing. Ph.D. thesis, Digital Enterprise Research Institute, National University of Ireland, Galwa (2013)
16. Prudhommeau, E., Carothers, G., Machina, L.: Rdf 1.1 turtle terse RDF triple language. W3C Recommendation, 25 February 2014
17. Tatbul, N., Çetintemel, U., Zdonik, S.B., Cherniack, M., Stonebraker, M.: Load shedding in a data stream manager. In: VLDB, pp. 309–320 (2003)
18. Jesper, H., Spyros, K.: High-performance distributed stream reasoning using S4. In: Ordering Workshop at ISWC (2011)
19. Le-Phuoc, D., Nguyen Mau Quoc, H., Le Van, C., Hauswirth, M.: Elastic and scalable processing of linked stream data in the cloud. In: Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J.X., Aroyo, L., Noy, N., Welty, C., Janowicz, K. (eds.) ISWC 2013, Part I. LNCS, vol. 8218, pp. 280–297. Springer, Heidelberg (2013)
20. Jain, N., Pozo, M., Chiky, R., Kazi-Aoul, Z.: Sampling semantic data stream: resolving overload and limited storage issues. In: Herawan, T., Deris, M.M., Abawajy, J. (eds.) Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013). LNEE, vol. 285, pp. 41–48. Springer, Heidelberg (2014). doi:10.1007/978-981-4585-18-7_5
21. Brian, B., Mayur, D., Rajeev, M.: Load shedding in data stream systems. In: Aggarwal, C.C. (ed.) Data Streams. ADS, pp. 127–147. Springer, Heidelberg (2007). http://www-cs-students.stanford.edu/datar/papers/mpds03.pdf
22. Agrawal, R., Imieliski, T., Swami, A.: Mining association rules between sets of items in large databases. ACM SIGMOD Rec. **22**(2), 207–216 (1993)
23. Hoan, Q., Mau, N., Le Phuoc, D.: An elastic and scalable spatiotemporal query processing for linked sensor data. In: Proceedings of the 11th International Conference on Semantic Systems. ACM (2015)
24. Belghaouti, F., Bouzeghoub, A., Kazi-Aoul, Z., Chiky, R.: Graph-oriented load-shedding for semantic data stream processing. In: 2015 International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM). IEEE, October 2015
25. Belghaouti, F., Bouzeghoub, A., Kazi-Aoul, Z., Chiky, R.: FreGraPaD: frequent graph patterns detection for semantic data streams. In: Tenth IEEE International Conference on Research Challenges in Information Science - RCIS (2016)
26. Dell'Aglio, D., Calbimonte, J.-P., Balduini, M., Corcho, O., Della Valle, E.: On correctness in RDF stream processor benchmarking. In: Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J.X., Aroyo, L., Noy, N., Welty, C., Janowicz, K. (eds.) ISWC 2013, Part II. LNCS, vol. 8219, pp. 326–342. Springer, Heidelberg (2013)
27. Tu, Y.-C., Liu, S., Prabhakar, S., Yao, B.: Load shedding in stream databases: a control-based approach. In: Proceedings of the 32nd International Conference on Very Large Data Bases, pp. 787–798. VLDB Endowment (2006)

Unsupervised Blocking of Imbalanced Datasets for Record Matching

Chenxiao Dou[1,2](✉), Daniel Sun[1,2], and Raymond K. Wong[1,2]

[1] School of Computer Science and Engineering,
University of New South Wales, Sydney, Australia
chenxiaod@cse.unsw.edu.au

[2] Data61, Commonwealth Scientific and Industrial Research Organisation (CSIRO),
ACT, Sydney, Australia

Abstract. Record matching in data engineering refers to searching for data records originating from same entities across different data sources. The solutions for record matching usually employ learning algorithms to train a classifier that labels record pairs as either matches or non-matches. In practice, the amount of non-matches typically far exceeds the amount of matches. This problem is so-called imbalance problem, which notoriously increases the difficulty of acquiring a representative dataset for classifier training. Various blocking techniques have been proposed to alleviate this problem, but most of them rely heavily on the effort of human experts. In this paper, we propose an unsupervised blocking method, which aims at automatic blocking. To demonstrate the effectiveness, we evaluated our method using real-world datasets. The results show that our method significantly outperforms other competitors.

Keywords: Record matching · Blocking · Imbalance · Heuristics

1 Introduction

A crucial step in integrating data from multiple sources is detecting and eliminating duplicate records [9]. This process is called Entity Matching, Record linkage, or Record Matching, which is a well known problem that arises in many applications such as address matching and citation de-duplication [3,5,6,16]. The goal of record matching is to identify records that represent the same real-world entities from a variety of data sources.

Machine learning has been playing an increasingly important role in the Record Matching problem. Some classical algorithms such as Support Vector Machine (SVM) and Decision Tree are adopted to predict whether a record pair is matched or not, based on the similarities between the entries [5,16]. However, selecting a training dataset is one of the most significant steps before learning process. To achieve a good classifier in terms of accuracy and recall, people have to label as many samples as possible, and consequently hiring human experts to hand-label the instances is too expensive in general.

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 172–186, 2016.
DOI: 10.1007/978-3-319-48743-4_14

In a real-world record matching task on a dataset with n records, it is obvious that the number of truly matched pairs is not larger than n. However, the total number of candidate pairs including matched pairs and non-matched pairs can reach $\theta(n^2)$. To reduce the number of candidate pairs, blocking techniques have been proposed to filter out the pairs that are unlikely matched [8,15].

The general idea is to divide the records into several blocks and only the records in the same blocks need to be compared. For the records in different blocks, it assumes that the pairs are unlikely to match. Take a dataset of academic publications for example. When detecting the duplicates, we may partition the dataset into small groups according to conference names. Papers published in different conferences cannot refer to the same real-world entities. To increase comparison opportunities for truly matched records, users may employ several blocking criteria to ensure that every duplicated pair is assigned into at least one block [18]. However, most of blocking criteria are designed manually and decided based on human experience.

In this paper, we propose a new blocking method that can automatically filter record pairs. For many imbalanced datasets, most candidate record pairs have low similarity. As a result, in a similarity space, areas resided by non-matched pairs have higher density than those with matched pairs. Therefore, unlike most blocking methods that take similarity thresholds as input parameters, our method uses density as the measure and requires two thresholds (to be discussed in detail later) of density provided by users. We target at matched and non-matched pairs that are entangled in a similarity space. Furthermore, to improve the efficiency of our method, we admit the monotonicity assumption on detecting the target region. Through our empirical studies, it is explicit that the property of monotonicity generally holds in real-world datasets. Our results demonstrate that the proposed method can significantly block non-matched pairs. In summary, the contributions of this paper are summarized as follows:

- Compared to manually blocking methods, our method is able to determine complex blocking criteria automatically.
- The performance of the proposed density based approach will not be affected as much by the selection of similarity functions.
- The monotonicity of datasets is examined and used to improve the efficiency of the proposed method.

The remainder of this paper is structured as follows. We discuss related work in Sect. 2. In Sect. 3, we present the background of this paper including the problem, the assumptions, and the definitions. Our proposed method is then presented in Sect. 4. After that, the proposed method is evaluated in Sect. 5. Finally, we conclude this paper in Sect. 6.

2 Related Work

Entity matching is a well-studied problem of determining whether two records refer to the same entity or not. To measure the similarity between records, a variety of similarity functions have been proposed. EditDistance, Jaccard similarity, Cosine similarity are three widely used measures for this problem [7]. These different similarity measures are the key criteria to identify truly matched records.

Many researchers have explored machine learning techniques [5,16] to solve the entity matching problem. Each similarity is regarded as a data feature, whose importance weight is automatically decided by a learning algorithm. But, there exists an imbalance issue that greatly affects the learning, that is, no learning algorithm can achieve an outstanding performance with an extremely imbalanced training set.

In order to reduce non-matched pairs, various blocking techniques are used to filter out the pairs that are unlikely matched. The traditional blocking criteria are manually designed according to the attributes of datasets [14]. Some learning algorithms are adopted to produce the blocking criteria automatically in [4,13]. Canopy Clustering [12] fast groups record pairs into overlapping sets with a loose threshold and then filters the non-matches with a tight threshold. An inappropriate blocking criterion may separate the truly matched records into different clusters. Most of the existing work use several blocking criteria to increase the chance of assigning the matched records to the same clusters. Whang et al. [18] proposed an iterative blocking framework, which enables record matching across different clusters. After that, they discussed how to update the existing blocking rules when new records are added in [17].

Most of the previous methods [3,4,6,12,13] use textual similarity as the main measure to filter pairs. In our work, we employ the density of pairs on a similarity space as the measure to find the non-matches. We also adopt the monotonicity assumption to improve the efficiency of our proposed density based method. The monotonicity assumption has also been introduced in [3,6] but they focused on the precision during learning, while in our approach, we focus on the property of monotonicity on the density during blocking.

3 Background

3.1 Problem Definition

In Table 1, the upper table contains three citation records from DBLP dataset, and the other has two records from ACM dataset. As shown in the tables, only the last records in the two tables are matched and the other three are non-matched. Thus, out of all 6 possible pairs between the two tables, there is only one matched pair, and this results in the imbalance problem. Given the entire datasets, the matched and the non-matched will be extremely unbalanced, and consequently from such datasets, it is difficult to sample a balanced training set, whose quality is important for the following learning process.

We denote the set of all record pairs as $\mathbb{C}$, the set of all matched pairs as $\mathbb{C}_{match}$, and the set of all non-matched pairs as $\mathbb{C}_{non}$. Our goal is to design a blocking method that could prune pairs in $\mathbb{C}_{non}$ and preserve pairs in $\mathbb{C}_{match}$ as many as possible in order to achieve a balanced data pool (the ideal imbalance

Table 1. Citation dataset

Title	Authors	Venue	Year
Safe query languages for constraint databases	Peter Z. Revesz	TODS	1998
Efficient filtering of XML documents for selective dissemination of information	Mehmet Altinel, Michael J. Franklin	Very large data bases	2000
Standards for databases on the grid	Susan Malaika, Andrew Eisenberg, Jim Melton	ACM SIGMOD record	2003
Title	Authors	Conf.	Year
Database techniques for the World-Wide Web: a survey	D. Florescu, A. Levy, A. Mendelzon	SIGMOD	1998
Standards for databases on the grid	S. Malaika, A. Eisenberg, J.	SIGMOD	2003

ratio is 1 : 1). The set of all blocked pairs is denoted as $\mathbb{B}$. Since it is difficult to prune the instances perfectly, the aim of our work is to make the output closely approximate the ideal result. It is measured based on three metrics: *Recall*, *Reduction Ratio* and *Imbalance Ratio*. We do not use *Precision* as a metric, because the aim of blocking is to balance the dataset other than classify the instances, and the high imbalance ratio will prevent any algorithm from directly finding the true matches.

$$Recall = 1 - \frac{\sum_{x \in \mathbb{C}_{match}} 1[x \in \mathbb{B}]}{|\mathbb{C}_{match}|} \tag{1}$$

$$Reduction\ Ratio = \frac{\sum_{x \in \mathbb{C}_{non}} 1[x \in \mathbb{B}]}{|\mathbb{C}_{non}|} \tag{2}$$

$$Imbalance\ Ratio = \frac{\sum_{x \in \mathbb{C}_{non}} 1[x \notin \mathbb{B}]}{\sum_{x \in \mathbb{C}_{match}} 1[x \notin \mathbb{B}]} \tag{3}$$

3.2 Similarity Space

Textual similarity is one of the main measures to decide whether two records are matched or not. Given a record pair $(r, s) \in R \times S$, we can use a variety of similarity functions to measure the similarity between attributes of R and attributes of S. Without loss of generality, we assume the returned scores from all the similarity functions are in $[0, 1]$. Given d different similarity measures, we can map a pair $(r, s) \in R \times S$ into a point $(s_1, \ldots, s_d) \in [0, 1]^d$ where s_i is the similarity score returned by the i-th similarity measure. We call $[0, 1]^d$ *similarity* *space*. Each dimension represents one similarity measure to evaluate a similarity between r and s.

In this paper, for the convenience of discussion, a similarity space is partitioned into regular cells. A *region* refers to a sub-space consisting of several consecutive cells and an *area* refers to an arbitrary sub-space in the similarity space.

3.3 Monotonicity

Monotonicity has been assumed for entity matching in [3,6]. For example, Arvind et al. [3] have studied the monotonicity of precision on improving the efficiency of their algorithm. In practice, the monotonicity assumption generally holds in many datasets. Table 1 is example records in two citation tables. Jaccard similarity on *Name* and EditDistance on *Title* are two frequently used similarity in Citation datasets. When we map the record pairs onto a 2D similarity panel shown in Fig. 1, we can observe that the dark area of lower similarity has much more points than the area of high similarity. Generally, in the similarity space $[0,1]^d$, the area of lower similarities may contain more points. We call this phenomenon as *Monotonicity of Density*. Intuitively, a pair of records with a lower textual similarity would have less possibilities to be matched. Thus, in this paper, we focus on how to use the monotonicity assumption to block record pairs.

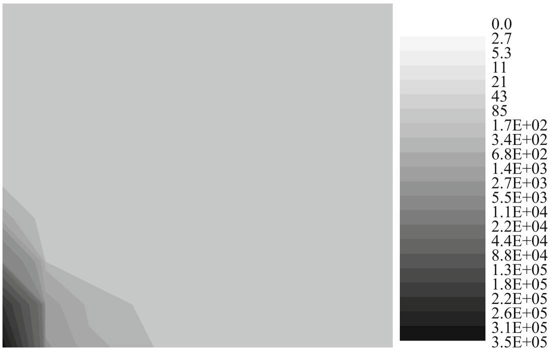


Fig. 1. Contour of citation pairs. The data are the publication records from DBLP and Google Scholar [1]. The panel on the left is the similarity space of two similarities. The right are the contour levels, which represent the numbers of points per unit area.

Prior to formally introducing *Monotonicity of Density*, we first give a formal definition of density. Due to the imbalance issue, the number of matched points is much smaller than that of non-matched ones after record pairs are mapped into $[0,1]^d$. So the density of non-matches around a given point is roughly defined as follows:

Definition 1. *Given a threshold r and a point f, the density of point f, denoted as $\rho(f)$, is defined as the number of points within a certain distance r to f.*

And then, we define the density of region as follow:

Definition 2. *In a similarity space* $[0, 1]^d$, *the density of a region is represented by the density of point* f *that has the lowest* $\rho(f)$ *among all points in the same region.*

We also define a partial ordering $\preceq$ on the points in a similarity space.

Definition 3. *Given two points* $f = (f_1, \ldots, f_d)$ *and* $f' = (f'_1, \ldots, f'_d)$, *if* $f_i \leq f'_i$ *for all* $1 \leq i \leq d$, *we say that* f' dominates f, *denoted as* $f \preceq f'$*; if* $f \preceq f'$ *and* $f_i \neq f'_i$ *for some* $1 \leq i \leq d$, *it is denoted as* $f \prec f'$.

Then, we formalize *Monotonicity of Density* as follows (Table 2):

Definition 4. *In a similarity space, for any two points* f *and* f' *such that* $f \preceq f'$, *if* $\rho(f) \geq \rho(f')$, *we say that the density is monotonic w.r.t.* $\preceq$.

Table 2. Notation table

Notation	Description
$\mathbb{C}$	The set of all pairs
$\mathbb{C}_{match}$	The set of matched pairs
$\mathbb{C}_{non}$	The set of non-matched pairs
$\mathbb{B}$	The set of blocked pairs
$\preceq, \prec$	The partial ordering (*dominate*) defined in Sect. 3
$\rho(f)$	The density of point f
d	The number of dimensions of similarity space
r	The distance threshold for computing density
T_1	The tight threshold
T_2	The loose threshold
k	The granularity parameter
p, f	The points
R, S	The datasets
$\mathbb{V}$	The set of granulated points
$\mathbb{M}$	The set of *MaxBound* points
$\mathbb{U}$	The set of *MinUnknown* points
$\mathbb{Z}$	The set of enumerated points with density no less than T_2

4 Proposed Method

4.1 Overview of Method

Motivated by the observation that non-matched pairs crowd greatly in the area of low similarity, we propose a heuristic method that can automatically block the non-matches. Our method consists of three main steps:

- **Preprocessing.** After mapping entity pairs into the similarity space, the number of points is rather huge. Due to the huge number, it is too expensive to compute the density of every points for finding the region of high density. To improve the efficiency, we split the similarity space into finite cells and only consider the corners of each cell as the candidate boundary points of high density region.
- **Search for the region with density no less than T_1.** The property of monotonicity notably holds in the region of rather low similarity. So in the second step, we use a *Binary Search* algorithm to find all sub-regions with density at least equal to T_1. The union of every sub-region is the target region where the density is at least equal to T_1. An example is shown in Fig. 2.
- **Search for the points with density no less than T_2.** In some region, the monotonicity may not hold strictly as well as in the region of density T_1, but there still exist many non-matches. Thus, in the rest of space, we run a recursive algorithm to enumerate the points of density no less than T_2, as shown in Fig. 2.

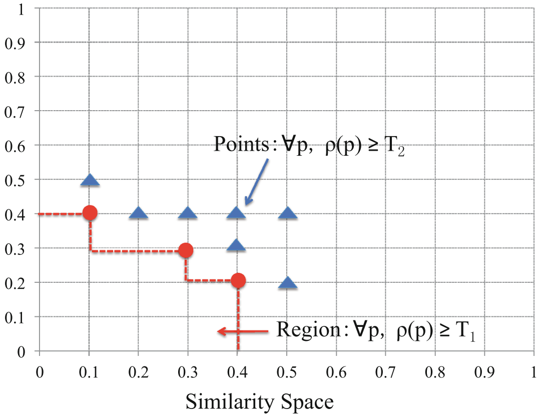


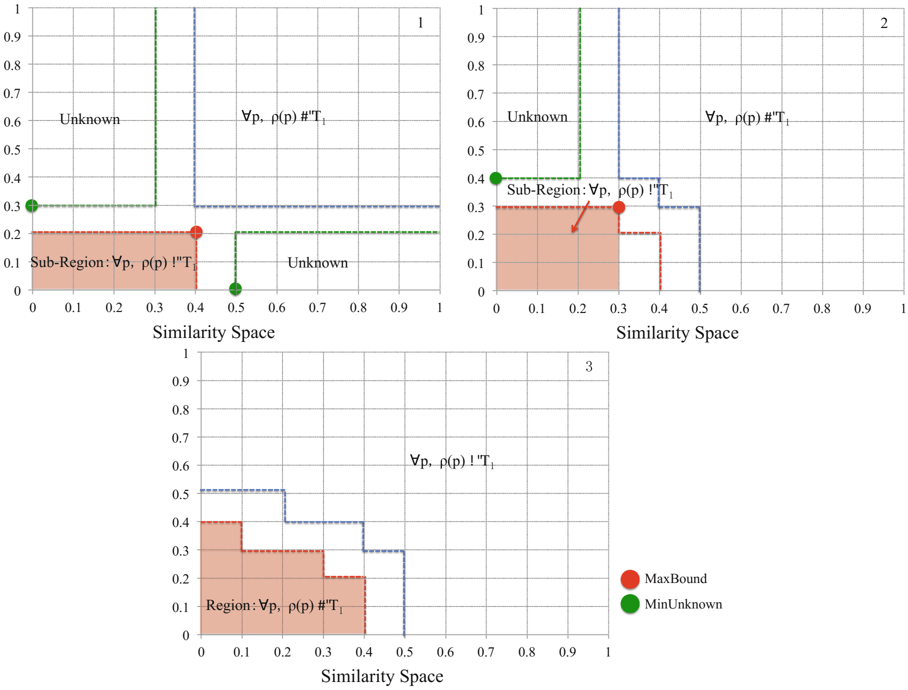
Fig. 2. Overview of the proposed method. The region with the density no less than T_1 will be blocked in the second step. The points with the density no less than T_2 will be blocked in the third step.

4.2 Preprocessing

In our method, the most expensive operation is to compute the defined density $\rho(f)$ of time complexity $\mathcal{O}(|\mathbb{C}|)$. As $\rho(f)$ is called frequently in the algorithms, we use an *R-tree* index[1] to mark the points and lower the complexity to $\mathcal{O}(\log(|\mathbb{C}|))$.

When searching the boundary points of high density region, it is infeasible to compute the density of every points in the similarity space. To improve the efficiency, we use an approximation technique that split the similarity space into finite cells and only check the points at the corners of each cell (Table 3).

Table 3. Search the region with the density no less than T_1.



In order to split the space, we fix an integer value k, called *granularity parameter*. Then we define a set $\mathbb{V}$ of points, any of which is in the form $(v_1, \ldots, v_d)$ where $v_i = j/k, j \in 0, 1, \ldots, k$. The set $\mathbb{V}$ partitions the similarity space into $(1+k)^d$ *cells*. And a *region* is an area formed by several consecutive cells. When considering the density of region, we only need to check the $(1+k)^d$ points in $\mathbb{V}$. The points are at the grid line crosses in Fig. 2.

[1] *R-tree* is a kind of tree data structures for indexing multi-dimensional information [10].

4.3 Search for the Region with Density No Less Than T_1

Now we show our solution of exploiting the monotonicity of density. Given two points $p \preceq p'$, if the monotonicity holds, $\rho(p) \geq \rho(p')$. From the practical experience in Fig. 1, this monotonicity generally applies in the region when the similarities are rather low. Considering the monotonicity of density, if we can find a point f with $\rho(f) \geq T_1$ and $\forall f' \succ f, \rho(f') < T_1$, this implies that the points f'' with $f'' \preceq f$ are all in the regions with the density no less than T_1. Formally, we define such point f as following:

Definition 5. *Given a threshold* T_1 *and* Monotonicity of Density*, we say a point* $p \in [0,1]^d$ *is maximally dense if* $\rho(p) \geq T_1$ *and* $\forall p' \succ p, \rho(p') < T_1$*. Such a point is called* MaxBound *point. And the set of such points is denoted as* $\mathbb{M}$*.*

According to the monotonicity, the point p that have the property $\exists p' \in \mathbb{M}, p \preceq p'$ should be in the region of density at least equal to T_1. In order to find the redundant non-matched points correctly, we set a relatively high threshold T_1 for enumerating the *MaxBound* points in $\mathbb{M}$. The detected record pairs, which have higher density than T_1, will be blocked out.

Next we propose our searching algorithm. The general idea is that when we locate one *MaxBound* point, it could always help divide the similarity space into three parts: unknown density, density at least equal to T_1 and density lower than T_1. In the unknown region, every point p has the property $\forall p' \in \mathbb{M}, p \not\preceq p'$ and $p \not\succ p'$. For the point $f \preceq \forall f'$ in one unknown region, we call such a point f as *MinUnknown* point, which has minimum density in this unknown region. And the set of all *MinUnknown* points is denoted as $\mathbb{U}$. To detect the density of unknown regions, our algorithm repeats the process of finding the *MaxBound* points on the remaining unknown regions.

Our algorithm starts at the point $(1/k, \ldots, 1/k)$ because it should be the first *MinUnknown* point in $\mathbb{U}$. Then we use *Binary Search* algorithm on every dimension to locate the *MaxBound* point. When finding a *MaxBound* point f, we know that the region resided by the points $p \preceq f$ has a density at least equal to T_1 and the region resided by the points $p \succ f$ has a density less than T_1. But the density of the remaining region is unknown. We put the point with minimum density of unknown region into $\mathbb{U}$ and start a new round of *Binary Search* on next *MinUnknown* point in $\mathbb{U}$. Repeat the same process until no region is unknown in $[0,1]^d$. We give a simple example on the 2-D similarity space in Fig. 2. In the first round, our algorithm starts at (0.1, 0.1). After the process of *Binary Searching*, the first found *MaxBound* point, $(0.4, 0.2)$, divides the space into four regions. The one resided by the *MaxBound* point is one sub-region of the target region of density T_1. The two unknown ones are the regions which need to be detected in next iteration. And the left one is the region with density less than T_1. In the second round, we start the searching algorithm from the two *MinUnknown* points founded in the last round, $(0, 0.3)$ and $(0.5, 0)$. After this round, the area of unknown regions is reduced and a new sub-region of density T_1 is formed as shown in the second step of Fig. 2. Then, we repeatedly detect the remaining unknown regions until the similarity space has been entirely explored.

The union region of all founded sub-regions is the target region with density no less than T_1.

As *MaxBound* points in $\mathbb{M}$ are on the boundary of region with density at least equal to T_1, we then enumerate each point $f \in \mathbb{M}$ and block all the points $p \preceq f$. According to *Monotonicity of Density*, the blocked points must have a density no less than T_1. For the second step, since we use *Binary Search* to locate the *MaxBound* points, the total number of points enumerated is $\mathcal{O}(\log k \cdot d \cdot |\mathbb{M}|)$.

```
Procedure EnumerateAllMaxBound($T_1$)
    $\mathbb{U} = \{(1/k, \ldots, 1/k)\}$
    foreach point $p \in \mathbb{U}$ do
        if $\rho(p) \geq T_1$ then
            $p_{max} \leftarrow$ FindMaxBound( $p, T_1$ )
            $\mathbb{M} \leftarrow \mathbb{M} \cup p_{max}$          /* $p_{max}$ is a MaxBound Point         */
            $\mathbb{U} \leftarrow$ UpdateMinUnknown( $\mathbb{U}, p_{max}$ )
        end
    end
Procedure UpdateMinUnknown($\mathbb{U}$, $p_{max}$)
    $\mathbb{U}_{new} \leftarrow \varnothing$
    foreach  point $p \in \mathbb{U}$ do
        if  $p \prec p_{max}$ then
            for $i \leftarrow 1, d$ do
                $p' \leftarrow p$              /* $p'$ is a MinUnknown point           */
                $p'[i] \leftarrow p_{max}[i] + 1/k$ /* $p'[i]$ is the $i$-th dimension of $p'$ */
                $\mathbb{U}_{new} \leftarrow \mathbb{U}_{new} \cup p'$
            end
        end
    end
Procedure FindMaxBound( $p$, $T_1$)
    for $i \leftarrow 1, d$ do
        $hi = 1$
        $lo = p_i$ while  $hi - lo \geq 2$ do
            $p[i] \leftarrow (hi + lo)/2$
            if  $\rho(p) \geq T_1$ then
                $lo = p[i]$
            else
                $hi = p[i]$
            end
        end
    end
    return $p$
```

Algorithm 1. Enumerate all *MaxBound* points

```
$\mathbb{Z} \leftarrow \mathbb{M}$
foreach point $p \in \mathbb{Z}$ do
    for $i \leftarrow 1, d$ do
        for $j \in \{-1, 1\}$ do
            $p' \leftarrow p$    /* $p'$ is a neighbour point of $p$ */
            $p'[i] \leftarrow p'[i] + j/k$    /* $p'[i]$ is the $i$-th dimension of $p'$ */
            if $p' \notin \mathbb{Z}$ and $\rho(p') \geq T_2$ then
                $\mathbb{Z} \leftarrow \mathbb{Z} \cup \{p'\}$
            end
        end
    end
end
```

Algorithm 2. Enumerate points with density no less than T_2

4.4 Search for the Points with Density No Less Than T_2

As mentioned above, the monotonicity property can coarsely distinguish points, but in some region distinguish-ability is not so strong. For a finer grain solution, the points in non-distinguishable region but with a high density should also be blocked out. In this section, we present a recursive algorithm to find the points.

In this step, we set another threshold $T_2 < T_1$ and search for the points of density no less than T_2. Our algorithm starts at some point p in the $\mathbb{M}$. If the neighbour point p' of p has $T_2 \leq \rho(p') < T_1$, we put it into an enumeration set $\mathbb{Z}$, which is equal to $\mathbb{M}$ at the very beginning. Then, we repeat this enumeration process at points in $\mathbb{Z}$, until no more eligible points are found and added into $\mathbb{Z}$. With assuming that the points close to a point $f \in \mathbb{Z}$ would also have a density at least equal to T_2, our blocking strategy is to remove all points within r to any one point in $\mathbb{Z}$.

The time consumption of this step is determined by T_2. The case that takes the longest time is when the density of each point in $\mathbb{V}$ on the rest space is just equal to T_2. When computing the density of point $p \in \mathbb{V}$, we need to count the number of points that are within a distance r to p. With some value of r, the density of p may count some points that are also counted by other $2d$ neighbour points of p in $\mathbb{V}$. To make the density identical, the number of shared points counted by two adjacent points in $\mathbb{V}$ should be same and equal to $T_2/2d$. After the last blocking step, the number of remaining points in $\mathbb{C}$, denoted as N_C, and the number of remaining points in $\mathbb{V}$, denoted as N_V, are known to users. There should exist $\frac{N_C}{T_2/2d}$ groups of points counted by two adjacent points in $\mathbb{V}$. And for each point, its density would be counted on $2d$ groups of shared points. Thus, if the maximal number of iterations does not exceed N_V, it can be represented as $\frac{N_C}{T_2/2d} \times \frac{2}{2d}$ equal to $\frac{2N_C}{T_2}$. The result shows the number of iterations does not matter with d. Even though the time complexity is $\mathcal{O}(\frac{N_C}{T_2})$, it should be noticed that N_C is much smaller than $|\mathbb{C}|$. The reason is that most of the non-matches are located in the region of high density and have been blocked in the last step.

5 Experiments

Our experiment were conducted on three datasets: *Restaurant*, *Goods* and *Citation*. *Restaurant* [1] is a collection of addresses of real-world restaurants. *Citation* [2] dataset contains a variety of publication records from DBLP and Google Scholar. *Goods* [2] is a dataset recording the product information on the websites of Google and Amazon. All of the datasets have two tables for record matching. The statistics of records in the datasets are shown in Table 4.

Table 4. Statistics of records

Datasets	Table A	Table B	Matched pairs
Restaurant	533	331	112
Citation	2294	2616	2224
Goods	1363	3266	1300

5.1 Method Comparison

In this section, we compare our method with *Standard Blocking* (SB) method [11] and *Canopies* (CANOPY) [12]. The two methods exploit different metrics for blocking. *Standard Blocking* clusters the records that share the same blocking keys into one block. In this experiment, we take the attributes of a dataset as the blocking keys. After having tried a variety of attribute combinations, we pick the one with good performance in terms of *Reduction Ratio* and *Recall* for comparison.

Canopies is an unsupervised clustering algorithm, which requires one loose similarity threshold θ_1 and one tight similarity threshold θ_2. It clusters the records that are textually similar to one block. From the similarity thresholds in the range $[0.3, 0.9]$, we choose the setting that achieves higher scores of *Reduction Ratio* and *Recall* for comparison.

Our method is a density-based clustering algorithm, which requires a loose threshold and a tight threshold of density. It blocks the records that are in high-density areas in the similarity space. We have four parameters all related to density computation. To simplify the process of tuning, we fixed $k = 20$ $r = 0.15$ and vary T_1, T_2 in $[100, 10000]$. In the following, the results are shown in Tables 5, 6, 7.

From the results, we can observe that the three methods all work well on *Citation* and *Restaurant*. The reason why there is no significant difference is that the two datasets have been cleaned, that is, they do not contain much noisy information. In such a scenario, similarity functions are highly credible, grading non-matched pairs a low score and fully duplicated pairs a high score. But for *Goods*, a dataset with noise, the two competitors do not work well. As the noise generally exists in the attributes of the dataset, the similarity scores

Table 5. Restaurant

	SB	CANOPY	OURS
Reduction ratio	0.847	0.924	0.992
Recall	1.000	1.000	0.982
Imbalance ratio	376	253	24

Table 6. Citation

	SB	CANOPY	OURS
Reduction ratio	0.824	0.944	0.986
Recall	1.000	0.999	0.994
Imbalance ratio	393	151	38

Table 7. Goods

	SB	CANOPY	OURS
Reduction ratio	0.831	0.997	0.996
Recall	0.274	0.377	0.885
Imbalance ratio	180	25	15

of truly matched data pairs may decrease to the level of non-matches'. Thus, a method targeting at the pairs with high similarity scores may not work anymore. As shown in Table 7, *Canopy* only captures 37.7 % matched pairs that reside in the area of high similarity.

Our method achieves an outstanding result when noise exists. As the algorithms are to search for the area of non-matches, the noise does not make an influence. For *Goods* dataset, the noise makes matched points enter into the low-similarity area, but also makes non-matched points move to the area with further low similarity. It is still possible to locate the area of non-matches, with using the property of density. Though some matches of low similarity cannot be found, our density-based method still achieves a high recall 0.885 on *Goods* dataset.

5.2 Evaluation on Density

As the density is defined as the number of instances in the area centered with r, the value of r plays an important role in the algorithms. A small r would make density more local and a big r is closer to the global density. In this section, we focus on studying the value of r.

Table 8. Restaurant

r	0.05	0.1	0.125	0.15	0.2
Reduction ratio	0.074	0.074	0.731	0.971	0.992
Recall	1.000	1.000	1.000	1.000	0.982
Imbalance ratio	3081	3081	895	96	24

Table 9. Citation

r	0.05	0.1	0.125	0.15	0.2
Reduction ratio	0.388	0.388	0.902	0.934	0.986
Recall	1.000	1.000	1.000	0.996	0.994
Imbalance ratio	1666	1666	266	178	38

Table 10. Goods

r	0.05	0.1	0.125	0.15	0.2
Reduction ratio	0.429	0.996	0.996	0.996	0.997
Recall	0.987	0.889	0.881	0.881	0.877
Imbalance ratio	1895	15	15	15	11

To study the effect of r, we fixed $k = 20, T_1 = 1000, T_2 = 200$ and vary the value of r in $\{0.05, 0.1, 0.125, 0.15, 0.20\}$. The experiment is conducted on three datasets as shown in Tables 8, 9, 10.

From the results, we can make two conclusions: First, with the increase of value of r, *Reduction Ratio* tends to be greater, while *Recall* and *Imbalance Ratio* tend to be smaller; Second, with a smaller r, our method does not have a good performance.

For the first conclusion, when r becomes greater, according to the definition of density, the thresholds T_1 and T_2 become relatively looser. For a point p with $\rho(p) \leq T_2$, after the increase of r, the area centered at p would become larger and include more points, making the new value returned by $\rho(p)$ possibly larger than T_2. Thus, with a loose threshold, there should be more pairs being blocked by the algorithm, and this leads to a higher *Reduction Ratio* and a lower *Recall.* The second conclusion is due to the noise. When enumerating the points with density no less than T_2, if the algorithm finds a point p with $\rho(p) < T_2$, the neighbour point $p' \succeq p, \rho(p') \geq T_2$ will have no chance to be enumerated. p is the noise point that may make our algorithm early stop, and is the reason why the results of $r = 0.05$ in all tables are bad. With a smaller r, the density has more chance to be affected by some noise points. But with a greater r, the effect of the noise would be balanced off, as shown in the experiment that the results of $r = 0.15, 0.2$ are all good.

6 Conclusion

In this paper, we studied the problem of blocking records for entity matching. Unlike the methods using similarity to filter entities, we used density as main measure for blocking. Our proposed method also exploits the monotonicity of density on improving the algorithm efficiency. Finally, we compared our density based approach with one similarity based approach and one index based approach. We evaluated our method on real datasets and demonstrate the superiority of our approach in terms of *Reduction Ratio* and *Recall.*

References

1. http://archive.ics.uci.edu/ml/
2. http://dbs.uni-leipzig.de/en/research/projects/object_matching
3. Arasu, A., Götz, M., Kaushik, R.: On active learning of record matching packages. In: Proceedings of ACM SIGMOD International Conference on Management of data, pp. 783–794. ACM (2010)
4. Bilenko, M., Kamath, B., Mooney, R.J.: Adaptive blocking: learning to scale up record linkage. In: 6th International Conference on Data Mining, ICDM 2006, pp. 87–96. IEEE (2006)
5. Bilenko, M., Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. In: KDD, pp. 39–48 (2003)
6. Chaudhuri, S., Chen, B.-C., Ganti, V., Kaushik, R.: Example-driven design of efficient record matching queries. In: Proceedings of 33rd International Conference on Very Large Data Bases, pp. 327–338. VLDB Endowment (2007)
7. Cohen, W.W.: Data integration using similarity joins and a word-based information representation language. ACM Trans. Inf. Syst. (TOIS) **18**(3), 288–321 (2000)
8. Dalvi, N.N., Rastogi, V., Dasgupta, A., Sarma, A.D., Sarlós, T.: Optimal hashing schemes for entity matching. In: WWW, pp. 295–306 (2013)
9. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate record detection: a survey. IEEE Trans. Knowl. Data Eng. **19**(1), 1–16 (2007)
10. Guttman, A.: R-trees: a dynamic index structure for spatial searching. ACM SIGMOD Rec. **14**, 47–57 (1984). ACM
11. Jaro, M.A.: Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. J. Am. Stat. Assoc. **84**(406), 414–420 (1989)
12. McCallum, A., Nigam, K., Ungar, L.H.: Efficient clustering of high-dimensional data sets with application to reference matching. In: Proceedings of 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 169–178. ACM (2000)
13. Michelson, M., Knoblock, C.A.: Learning blocking schemes for record linkage. In: Proceedings of National Conference on Artificial Intelligence, vol. 21, p. 440. AAAI Press, MIT Press, Menlo Park, London (2006) (1999)
14. Newcombe, H.B.: Handbook of Record Linkage: Methods for Health and Statistical Studies, Administration, and Business. Oxford University Press Inc., Oxford (1988)
15. Shu, L., Chen, A., Xiong, M., Meng, W.: Efficient spectral neighborhood blocking for entity resolution. In: IEEE 27th International Conference on Data Engineering (ICDE), pp. 1067–1078. IEEE (2011)
16. Tejada, S., Knoblock, C.A., Minton, S.: Learning domain-independent string transformation weights for high accuracy object identification. In: Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 350–359. ACM (2002)
17. Whang, S.E., Garcia-Molina, H.: Incremental entity resolution on rules and data. VLDB J. **23**(1), 77–102 (2014)
18. Whang, S.E., Menestrina, D., Koutrika, G., Theobald, M., Garcia-Molina, H.: Entity resolution with iterative blocking. In: SIGMOD Conference, pp. 219–232 (2009)

Partially Decompressing Binary Interpolative Coding for Fast Query Processing

Xi Fu[1,2], Peng Li[1](✉), Rui Li[1], and Bin Wang[1,2]

[1] Institute of Information Engineering,
Chinese Academy of Sciences, Beijing, China
{fuxi,lipeng,lirui,wangbin}@iie.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China

Abstract. Inverted index is the core data structure in large scale information retrieval systems such as Web search engine. Index compression techniques are usually used to reduce the storage and transmission time from disk to memory. Many index compression schemes have been proposed and among them Binary Interpolative Coding (IPC) is one of the most widely used schemes due to its superior compression ratio (CR). However, the decompression speed of IPC is relatively slow, thus fully decompressing (FD) IPC will slow down the whole process of online query processing. In this paper, we first point out that it is unnecessary to fully decompress all the IPC nodes in query processing and then propose a partial decompression (PD) algorithm for IPC. Experimental results on two publicly available standard corpora show that compared with normal IPC our algorithm performs 40 % faster for Boolean conjunctive queries and 20 % faster for Rank queries without additional storage consumption.

Keywords: Index compression · Binary interpolative coding · Inverted file

1 Introduction

Information retrieval systems have been widely used in many online applications including search engine, library system and e-commerce system etc. To speed up the retrieving process an important data structure named inverted index file (IF) has been proposed. In general, IF needs to be read from disk into memory during online processing, however, the uncompressed inverted index is almost as large as original collection [1]. Therefore, lots of compression schemes have been proposed [2–8]. In general, schemes aiming at high compression ratio (CR) usually have a slower decompressing speed, so most methods try to make a tradeoff between memory occupation and decompression speed. Binary interpolative coding (IPC) [8] is one of those schemes with high CR while its decompressing speed is relatively slower.

Because of its high CR, IPC can be used in those extremely high CR required situation. For example, when compressed *IF* is larger than the internal memory space, high CR schemes help to maintain more cache in internal memory and also reduce the amount of data read from external memory. For instance in [13] IPC is used to compress the huge versioned document collections. However, the relatively heavy time consummation during online query processing limits its further use. Any improvement

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 187–195, 2016.
DOI: 10.1007/978-3-319-48743-4_15

of its processing speed is not trivial. Many researchers have realized acceleration of online processing is meaningful and proposed some improvements [9, 10]. Most of these improvements use extra space to exchange with time which weakens the advantage of high CR. Our work aims at exploring the acceleration potential of query processing when *IF* is compressed in IPC form while without additional space consummation.

We analyze its decompressing procedure in detail and find that not every node needs to be decompressed during query processing. Therefore, we propose a PD algorithm. We will describe our algorithm in detail and finally validate its efficiency.

Our contributions are double fold. First, we propose a partial decompression (PD) algorithm to process Boolean conjunctive query and Rank query when *IF* is compressed in IPC form. Second, we experimentally validate its efficiency on two publicly available corpora.

This paper is organized as follows. We briefly introduce the IPC and background in Sect. 2. In Sect. 3 we will explain our algorithm. We will check its efficiency by experiment in Sect. 4. Finally, we give our conclusion.

2 Background and Related Work

2.1 Background

An inverted index file (*IF*) consists of a term dictionary and an inverted list (also called posting list) for each term; we mainly focus on the decompression of inverted lists in this paper.

A term t and its inverted list can be described as $\{f_t : d_{t,1}, d_{t,2}, d_{t,3} \ldots d_{t,f_t}\}$, where f_t is the total number of documents containing the term t and $\mathrm{d}_{\mathrm{t},1}, \mathrm{d}_{\mathrm{t},2}, \mathrm{d}_{\mathrm{t},3} \ldots \mathrm{d}_{\mathrm{t},\mathrm{f}_\mathrm{t}}$ are IDs of them which are usually sorted in increasing order.

IF is a convenient tool to process queries, however, *IF* is almost as large as original document file so storing them simply will need too much space. Therefore, it needs compression before being read into memory. Until now many compression methods are proposed. Some methods are designed for delta values [3–5]. Among them OPTPFD [7] is competitive both in time and space so widely used in SE system. We use OPTPFD as one of the baseline methods.

Methods designed for delta values utilize the distribution of deviation; its efficiency relies on some assumption of distribution. Another type of methods [2, 6, 7] designed for the original value utilizes the property of monotonously increasing list. Among them EF [6] is another competitive method and will be used as another baseline method in this paper. Partitioned-EF [7] is the improvement of EF and can be regard as one of the state-of-the-art methods.

Although many coding have been proposed, IPC is still competitive for its incomparable CR. We will check its detail in next section.

2.2 Binary Interpolative Coding and Its Improvements

Binary Interpolative Coding (IPC) [8] is a useful coding for its high CR. IPC is designed for original values which numbers are listed in strictly increasing order. Its idea is limiting each number by its order and interval to minimize its entropy. Use the classic example on [8], the inverted list is <7:3, 8, 9, 11, 12, 13, 17> with a range of [1, 20]. Firstly we encode the middle number 11, which have 3 numbers on either side. Its range is [1 + 3, 20 − 3] with a length of 17 − 4 + 1 = 14, so it needs 4 bits and is encoded to 11 − 4 = 7, 0111. The next number is the middle number in the left part, which is 8 with one number on either side. Its range is [1 + 1, 10 − 1] with a length of 9 − 2 + 1 = 8, so it needs 3 bits and is encoded to 8 − 2 = 6, 110, etc. It is a recursive procedure as a binary tree which is called IPC tree. See Fig. 1. Finally we record it in pre-order traversal order (also called PLR order: parents, leftchild, rightchild). The above list is recorded as: 0111, 110, 010, 0, 000, 011. Note the commas do not need to be stored, because length of each node is already determined by its parent nodes.

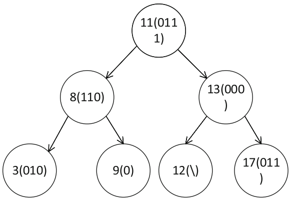


Fig. 1. The encoding process tree of IPC

Decompressing process will reverse the process. To restore the list we need computations about many messages of each node, which are expensive and will slow down the query processing. Researchers have realized to accelerate processing time is helpful and proposed some improvements. To our best knowledge the following two papers are found. In 2004 Cheng proposed a unique-order IPC [9]. They cut *IF* into fixed length of pieces and claimed at least two advantages would be taken. With fixed length *g* of IPC pieces the decoding process can be accelerated by pre-computer order, another advantage is that skipping is feasible during intersection. However, parameter *g* is difficult to determine because the length of list varies from few to million. In 2010 Teuhola proposed a log-time IPC [10]. Their idea is to compute the longest possible length of an interval with a particular number of elements, then encoding that part of list with its longest possible length. By this it is easy to locate a particular element for skipping and random access. However it requires additional space and its space occupation might be larger than OPTPFD from the report. These two improvements try to accelerate decompressing of IPC by using additional space so that the advantage of high CR has been weakened. Our work aims at developing the acceleration potential without reducing its CR (do not change its structure).

3 PD and FD in Query Processing

3.1 Analyze of FD Process in Conjunctive Boolean Query Processing

Firstly we introduce the algorithm of FD. We need to rebuild the binary tree in Fig. 1 during FD. Because data in IPC is arranged in PLR order, the rebuilding procedure also proceeds in PLR order. For each node we need to compute the information of its *left* (number of nodes on the left sub-tree), *right*, *max*, *min* and *len* (length of bits read from *IF*) by parents' values, next we read *len* bits from *IF* to get the *tmpvalue*, then we can restore this node and its children. Here we proposed an algorithm using stack to decompress IPC tree. We need a stack *S* of structure contains the necessary message of each node including the fields mentioned above. We unitize the algorithm of in-order traversing of binary tree with stack and restore each value recursively. Each value can be computed by its parents' nodes. A global point *CurPointer* is initialized as the start address of the according list in *IF*. Finally we put original values into a list in order, which is just the inverted list. Next work is intersecting all those restored lists. We start at the shortest one so the intersection list is always short. However in experiment we find the processing of FD occupies at least 90 % of the total processing time which means decompression needs acceleration.

Note the structure of IPC coding is a binary sort tree which might apply skipping operations in conjunctive Boolean query. Refer to [12] we get following two conclusions: First, skip list can accelerate conjunctive query processing. Second, most of the Boolean queries involve only conjunction operations. So we also want to apply skipping to implement partial decompression in IPC tree which however cannot be used as skip list directly. That is because in a skip list a skipped node should have an additional pointer point to the next skipped node. Besides, values in IPC tree are not stored in order. However we have Observation 3.1 which makes skipping possible.

Definition 3.1: For any sub-tree in IPC tree with a root of *R*, there is a path from root to leaf generated by each time travels the left branch. Then we reverse the path, nodes on the reversed path can be regarded as a list and defined as **left list**.

Observation 3.1: Values of nodes on *left list* in any sub-tree are sorted in increasing order. For each node on *left list*, all the values of nodes on its right sub-tree are between the value of current node and next node on *left list* (or the overall maximal).

Observation 3.1 can be easily explained by the property of binary sort tree. By Observation 3.1 IPC tree can be regarded as recursive levels of skip list. Consider the same example in Sect. 2.2, its IPC tree is showed in Fig. 1. We could also change it into a multilayer skip list by rotating the tree clockwise till its *left list* to be horizontal. We could find that the nodes 3, 8, 11 could be viewed as skipped nodes in skip list. Skipped nodes and nodes in its right sub-tree could be viewed as an inner block. But not like skip list IPC coding use pre-order to store data. For a tree in IPC it is stored by two parts including nodes on *left list* which will be stored by the reversed *left list* order and right sub-tree of each node on *left list* which will be stored by *left list* order. For example in Fig. 1 the storing order is {11, 8, 3, {9}, {13, 12, 17}}.

Consider the compression order we know after having accessed the root node we could easily decompress the entire *left list* without decompressing other nodes.

However to locate the start point of a right sub-tree of a particular node *N* we have to compute the stored length of right sub-tree of all nodes which are before *N* in the *left list*. For instance if we have a list of {1, 13} to intersect with IPC tree in Fig. 1, after we restored the *left list* and finish comparison we decide to skip the right sub-tree of node 3 and 8, we need to know the length of sub-tree {9} then we can locate the second sub-tree on *IF*. It seems computation on sub-tree is inevitable. However we find the computation of the length of a sub-tree is faster than restoring the total tree.

Definition 3.2: During query processing a tree compressed in IPC, if we do not restore all the values of nodes on the tree, instead we just compute its length in *IF*, we call this operation as **skipping the tree**. Nodes on that tree are called **skipped nodes**.

Observation 3.2: Skipping a tree is faster than restoring (or decompressing).

The reasons are as follows. First, to skip a node we just need to know its *left, right, len* and *tmpvalue*. Next, computations of *max*, *min* and original value usually involve with operation of big integers. Last, skipping operation in leaf nodes, which have a large amount in a tree, is so fast that do not even need to allocate an addition node. By experiment we find skipping a large tree costs almost 3/4 time of restoring.

Another noticeable advantage of PD is sometimes we can terminate the traversal of tree earlier. For example we have a list of {1, 2} and intersect it with the tree in Fig. 1, after we restore the *left list* of root node we can safely ignore remain ones.

In conclusion, the normal processing of IPC in conjunctive query is fully decompressing the inverted list and then doing intersection while our idea is doing decompressing with intersection simultaneously. We decompress only the nodes which are possible to be a result and skip (or ignore) the others.

3.2 PD Process in Conjunctive Boolean Query and Rank Query Processing

We check Boolean query first. Firstly we describe our skipping function to skip a sub-tree with a particular root. Skipping function is similar with restoring function and shares the stack. We need only the message of *left, right, length* and *tmpvalue*, which consist a structure of skipped node. When the skipped node is a leaf, we just need to move the pointer on *IF* while without restoring the node. Next we check the timing for executing skipping. We denote the restored list as L and list compressed in IPC as T. For a node in the *left* list of a particular node in T, we denote the maximal of range of its right sub-tree as *m* (*m* = *max* + *right*). When the current value in L is larger than *m*, the total right sub-tree can be skipped. See Fig. 2 for example.

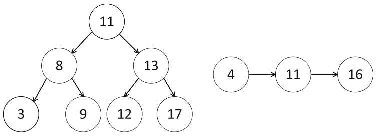


Fig. 2. An example of skipping

The left tree is compressed in IPC while the right one is totally restored. First we decompress the *left list* of root node in the IPC tree, which is {3, 8, 11}. Next we process the intersection of *left list* and the restored list. When we process at the node 8 on IPC, the current pointer on the restored list stays at the node 11. We find value *m* of node 8 is 10 and less than 11, so we could skipped the right sub-tree (*right skip*) of node 8. When we process at node 13, the current pointer on the restored list stays at the node 16. We can safely skip the left sub-tree (left *skip*) of node 13. In algorithm for each outer loop left skip may occur only when start the outer loop and before restoring the left edge. Here is the algorithm. Let *L* be the list totally restored, *T* be the *IF* and *Q* be the intersection result array. *Lcur* and *Qcur* are current pointers on L and Q separately and initialized as 0.

```
Restore root node of T and push into stack S
while(Tcur=S[top]!=null)
  if(L[Lcur]>=root->orgvalue)
    Skip left child of root //execute the left skip
  else
    Restore left list of sub-tree with the root Tcur from T and push them by decompression order into S
    while(true)  //loop for array L
      if(L[Lcur]<=Tcur->orgvalue)
        if (L[Lcur]==Tcur->orgvalue)
          Q[Qcur++]=L[Lcur]  //copy the node into result list
      else if(Lcur<maxlength) //maxlength of array L
        Lcur++
        continue
        else
        return //array L have reached the end and ignore the remain nodes in T
      else
        pop S
        if((L[Lcur]<(Tcur->rightvalue+Tcur->maxvalue))&& Tcur->right>0) //need to check its right child
          restore right child of temp and push it into S
          break //break into the outer loop

        skip   right   child   of   temp
        //execute the right skip
```

Next work is simple. We choose the shortest inverted list and totally decompress it. Then use it as *L* to intersect with following list to get the result list of query.

Rank queries are similar. Although rank queries are more like disjunctive Boolean query we can use the *continue* strategy in [12] to execute skipping when processing the ID list. After having processed the ID list we will record the position of the result of intersection into a list. We can utilize this list to access the according values on frequency IPC tree and skip the unnecessary nodes.

4 Experiments

We process experiments on the following corpus. PPD (PeopleDaily91–93) is a collection of short news reported by Chinese official news agency: NCNA (New China News Agency). It contains 135,193 documents and 3,171,009 terms. Gov2 is the TREC 2004 Terabyte Track test collection which includes millions of documents crawled from.gov website. Gov2 are mainly in English and with 9,122,147 documents and 164,443,902 terms. For data in Chinese we use *ansj* (http://www.ansj.org/) for Chinese word segmentation, we do not need stemming technology and remove stop word. For data in English we use Porter2 stemmer and have removed stop word. The experiment machine has a CPU of Inter Core i5-2400, main memory of 4G and WIN7 32 OS. All online processing programs are coding in C++ and we use g++ 4.3 for complier. Queries are randomly chosen from segments of some documents.

Next we will check the efficiency of PD during query processing. Our space consummation remains the same with PD and we focus on its time consummation.

Above Tables show the result of PD compared with FD, PFD and Elias-Fano coding. Actually in practical PD will perform better than above because our programs are all running in inner memory and the advantage of less amount of data transmission between inner and external storage have not been showed.

From Table 1 we find in Boolean conjunctive query PD is comparable with that of Elias-Fano. From Table 2 we know PD in Rank query do not performed as well as Boolean conjunctive query. We find in large corpus PD will take more advantage, but even in Gov2 PD is about a quarter slower compared with the popular coding OPTPFD. That partly because more lists needs to be totally restored in Rank query.

Table 1. Average running time of Boolean conjunctive query

Length of query	FD per query ave time ($\bar{t}$) (ms)		PD ($\bar{t}$) (ms)		OPTPFD [7] ($\bar{t}$) (ms)		Elias-Fano [8] ($\bar{t}$) (ms)	
	PPD	Gov2	PPD	Gov2	PPD	Gov2	PPD	Gov2
2	0.244	1.502	0.196	1.161	0.164	1.064	0.172	1.069
4	0.475	2.997	0.330	2.047	0.290	1.858	0.325	1.954
6	0.663	4.288	0.448	2.453	0.379	2.504	0.431	2.667
8	0.875	5.474	0.596	3.399	0.498	3.186	0.569	3.372
10	1.079	6.561	0.728	4.160	0.619	3.812	0.699	4.015

Table 2. Average running time of Rank query (Acu = 1000 on PPD and 5000 on Gov2)

Length of query	FD Per query ave time ($\bar{t}$) (ms)		PD ($\bar{t}$) (ms)		OPTPFD [7] ($\bar{t}$) (ms)		Elias-Fano [8] ($\bar{t}$) (ms)	
	PPD	Gov2	PPD	Gov2	PPD	Gov2	PPD	Gov2
2	0.6551	6.7909	0.6568	6.7990	0.4092	4.5840	0.5267	5.6436
4	1.0104	11.6633	0.9539	11.100	6.5575	8.494	7.633	9.307
6	1.6994	19.4289	1.5144	17.117	1.0978	15.1139	1.2439	15.183
8	2.4501	28.1508	2.1775	23.957	1.513	18.917	1.7984	21.957
10	3.1960	38.9273	2.8412	32.459	2.472	26.120	2.612	31.268

5 Conclusion

IPC is an effective coding in CR with a relatively slower processing speed, any improvement on its processing speed is not trivial. We proposed a partially decompression algorithm which can accelerate query processing speed. We try to reduce the time consummation by saving any computation unnecessary and we prove our method is useful. Our algorithm is about 40 % faster for Boolean conjunctive query and 20 % faster for Rank query without additional storage consumption compared with FD.

Acknowledgments. We would like to thank the anonymous reviewers for their valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China (grant No. 61402466, 61572494), the Strategic Priority Research Program of the Chinese Academy of Sciences (grant No. XDA06030200), and the National Key Technology R&D Program (grant No. 2012BAH46B03).

References

1. Zobel, J., Moffat, A.: Adding compression to a full-text retrieval system. Softw. Pract. Exp. **25**(8), 891–903 (1995)
2. Golomb, S.W.: Run-length encodings. IEEE Trans. Inf. Theor. **12**(3), 399–401 (1966)
3. Williams, H.E., Zobel, J.: Compressing integers for fast file access. Comput. J. **42**(3), 193–201 (1999)
4. Zukowski, M., Heman, S., Nes, N., Boncz, P.: Superscalar RAM-CPU cache compression. In: Proceedings of the 22nd International Conference on Data Engineering (ICDE), no. 59, pp. 1–12. IEEE (2006)
5. Yan, H., Ding, S., Suel, T.: Inverted index compression and query processing with optimized document ordering. In: Proceedings of the 18th International Conference on World Wide Web (WWW), pp. 401–410. ACM (2009)
6. Vigna, S.: Quasi-succinct indices. In: Proceedings of the 6th International Conference on Web Search and Data Mining (WSDM), pp. 83–92. ACM (2013)
7. Ottaviano, G., Venturini, R.: Partitioned Elias-Fano indexes. In: Sigir (2014)
8. Moffat, A., Stuiver, L.: Binary interpolative coding for effective index compression. Inf. Retrieval **3**(1), 25–47 (2000)

9. Cheng, C.S., Shann, J.J.J., Chung, C.P.: Unique-order interpolative coding for fast querying and space-efficient indexing in information retrieval systems. Inf. Process. Manag. **42**(2), 407–428 (2006)
10. Teuhola, J.: Interpolative coding of integer sequences supporting log-time random access. Inf. Process. Manag. **47**, 742–761 (2011)
11. Anh, V.N., Moffat, A.: Pruned query evaluation using pre-computed impacts. In: SIGIR, pp. 372–379. ACM, New York (2006)
12. Moffat, A., Zobel, J.: Self-indexing inverted files for fast text retrieval. Systems **14**(4), 349–379 (1996)
13. He, J., Yan, H., Suel, T.: Compact full-text indexing of versioned document collections. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, 02–06 November, Hong Kong, China (2009)

Using Changesets for Incremental Maintenance of Linkset Views

Vânia M.P. Vidal[1](✉), Marco A. Casanova[2], Elisa S. Menendez[2], Narciso Arruda[1], Valeria M. Pequeno[3], and Luiz A. Paes Leme[4]

[1] Federal University of Ceará, Fortaleza, CE, Brazil
{vvidal,narciso}@lia.ufc.br
[2] Department of Informatics, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, RJ, Brazil
casanova@inf.puc-rio.br, elisasmenendez@gmail.com
[3] INESC-ID, Porto Salvo, Portugal
vmp@inesc-id.pt
[4] Fluminense Federal University, Niteroi, RJ, Brazil
lapaesleme@ic.uff.br

Abstract. In the Linked Data field, data publishers frequently materialize linksets between two datasets using link discovery tools. However, when the datasets are continually updated, a materialized linkset must also be updated since the links may no longer meet the linkage rules. To help solve this problem, this paper presents an approach for maintaining linksets, which treats linksets as materialized views, is based on changesets and adopts an incremental strategy. The paper formalizes the materialized linkset maintenance problem based on changesets and indicates that our approach correctly maintains materialized linksets views. Finally, it suggests an architecture and describes an implementation and experiments to validate the proposed approach.

Keywords: RDF dataset interlinking · Linked data · View maintenance

1 Introduction

The Linked Data initiative [1] defines best practices for publishing and interlinking data on the Web using RDF triples to represent the data. Briefly, a *dataset* is simply a set of RDF triples. A *link* is an RDF triple of the form (s, p, o), where s and o are resources defined in two distinct datasets. A *linkset* is a set of links.

Link discovery tools help create and materialize linksets. These tools are typically semi-automatic in the sense that users have to define a set of *linkage rules* that specify conditions that resources must fulfill to be interlinked. However, when datasets are continually updated, the maintenance of a materialized linkset requires attention since the links may no longer meet the linkage rules that originated the linkset. To inform consumers about changes, an RDF dataset should publish *changesets* [3] to indicate the difference between two states of the dataset.

In this paper, we present an approach for maintaining materialized linksets. The approach we propose: (1) treats linksets as materialized views, called *linkset views*;

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 196–204, 2016.
DOI: 10.1007/978-3-319-48743-4_16

(2) accounts for the facts that a linkset is computed by (complex) linkage rules and that the linkset does not contain the property values used by the linkage rules; (3) uses the changesets published by the source datasets to compute the changes that must be applied to a materialized linkset to keep it consistent with the new states of the source datasets; (4) adopts an incremental strategy. The proposed approach has two main steps. The first step uses the changesets, published by the source datasets, to compute the set of updated resources that are relevant to the materialized linkset. The second step updates the links for the relevant resources.

The contributions of the paper are: (i) we formalize the materialized linkset maintenance problem based on changesets; (ii) we define an approach that uses changesets to incrementally maintain materialized linksets and informally illustrate how it works; (iii) we provide two theorems that indicate that the proposed algorithms correctly maintains materialized linksets views; (iv) we describe an implementation and experiments to validate the approach.

Several tools were designed to create linksets [4, 5, 9]. The introduction of views, as suggested in [2], would simplify the configuration of the tools designed to create links. In another direction, tools, such as DSNotify [6], were designed to inform database administrators about dataset changes and to allow them to preserve link integrity. The proposed approach is based on, but not reducible to such incremental view maintenance strategies. Indeed, a linkset is not a regular view computed by querying two datasets, but it is created using linkage rules that frequently involve computing entity similarity. Furthermore, a linkset does not contain the property values that the linkage rules use. Endris et al. [3] presented a framework for interest-based RDF update propagation that can consistently maintain a full or partial replication of large LOD datasets. This framework is also based on changesets, but the solution can only be applied when the view mappings are direct mappings. The approach proposed in this paper goes further and considers linkset views defined by expressive mappings.

The paper is organized as follows. Section 2 introduces basic definitions and a running example. Section 3 presents our approach for maintaining linksets views. Section 4 describes an implementation and experiments to validate the proposed approach. Finally, Sect. 5 contains the conclusions.

2 Linkset Views

2.1 Linkset View Definition

To make the paper self-contained, we introduce an abstract notation to define *catalogue views* and *linkset views* with the help of mapping rules [7]. In the rest of this paper, $\sigma_S(t)$ denotes the state of S in time t, where S can be a source dataset or a view, and M $[\sigma_S(t)]$ denotes the set of triples defined by a set M of mapping rules against $\sigma_S(t)$.

A *catalogue view definition* is a triple $\mathbf{V} = (V_V, S_V, M_V)$, where

- V_V is the vocabulary of $\mathbf{V}$, also called the *view vocabulary*, and consists of a single class and a set of datatype properties
- S_V is the source dataset which exports the view $\mathbf{V}$, described by a vocabulary V_S

- M_V is a set of mapping rules that map concepts of V_S to concepts of V_V, called the *view mapping.*

A *materialization* of view **V** at time *t* is obtained by computing $M_V[\sigma_{S_V}(t)]$ and storing it as part of a dataset.

A *linkset view definition* is a quintuple **L** = (*P, V*$_{\mathbf{L}}$, **F**, **G**, μ), where

- *P* is an object property
- $V_{\mathbf{L}}$ is the *match vocabulary* of ***L*** and consists of a single class and a set of datatype properties
- **F=** *(*$V_{\mathbf{F}}$*,* $S_{\mathbf{F}}$*,* $M_{\mathbf{F}}$*)* and **G=** *(*$V_{\mathbf{G}}$*,* $S_{\mathbf{G}}$*,* $M_{\mathbf{G}}$*)* are catalogue view definitions where
- $V_{\mathbf{F}} = V_{\mathbf{G}} = V_{\mathbf{L}}$. Thus, $V_{\mathbf{L}}$ is the common vocabulary for exported views **F** and **G**
- *μ* is a *2n*-relation, called the *match predicate* of **L.**

Let $V_{\mathbf{L}} = \{C, P_1,\ldots,P_n\}$ be the match vocabulary of **L**. Let $\sigma_{\mathbf{F}}(t)$ and $\sigma_{\mathbf{G}}(t)$ be states respectively of **F** and **G** in time *t*. The *state* of **L** in time *t* is the set $\sigma_{\mathbf{L}}(t)$ defined as: *(s, p,o)* $\in \sigma_{\mathbf{L}}(t)$ iff there are triples *(s, rdf:type, C), (s,* P_1*,* s_1*),…, (s,* P_n*,* s_n*)* $\in \sigma_{\mathbf{F}}(t)$ and *(o, rdf:type, C), (o,* P_1*,* o_1*),…, (o,* P_n*,* o_n*)* $\in \sigma_{\mathbf{G}}(t)$ such that $(s_1,\ldots, s_n, o_1,\ldots, o_n) \in \mu$

2.2 Running Example

In this section, we illustrate how to define a linkset view. Consider the *MusicBrainz* (http://musicbrainz.org/doc/about) dataset, which uses the Music ontology. Figure 1 shows a fragment of the Music ontology, which reuses terms from three well-known vocabularies: *FOAF* (Friend of a Friend), *MO* (Music Ontology) and *DC* (Dublin Core). Consider the DBpedia (http://wiki.dbpedia.org/about) dataset, which uses the *DBpedia* Ontology *(dbo)*. Figure 2 shows a fragment of *DBpedia* ontology.

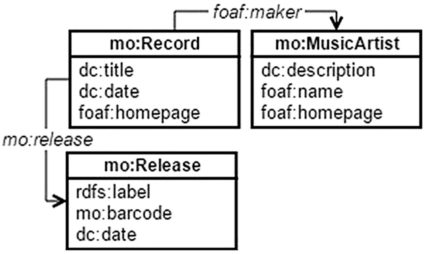


Fig. 1. A fragment of the *Music Ontology*

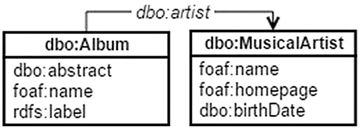


Fig. 2. A fragment of the *DBpedia Ontology*

Suppose that a user wants to create *sameAs links* between instances of the class *Record* in the *MusicBrainz* dataset and instances of the class *Album* in the *DBpedia* dataset. For this purpose, the user creates the linkset view definition **L** = (*owl:sameAs*, $V_{\mathbf{L}}$, **F**, **G**, μ), where: $V_{\mathbf{L}}$={*mo:Record, dc:title, mvl:artistName, dbo:releaseDate*}; the vocabulary $V_{\mathbf{L}}$ reuses terms from *DBpedia and Music Ontologies* and defines a new term *mvl:artistName*; **F** and **G** are catalogues views exported by *DBpedia* and *MusicBrainz*, respectively, with mapping rules $M_{\mathbf{F}}$ and $M_{\mathbf{G}}$ from *DBpedia and Music Ontology* to common vocabulary $V_{\mathbf{L}}$, respectively:

$M_{\mathbf{F}}$: *mo:Record(x)* ← *dbo:Album(x)*
dc:title(x, y) ← *dbo:Album(x); foaf:name(x, y)*
mvl:artistName(x, y) ← *dbo:Album(x); dbo:artist(x, z); foaf:name(z, y)*
dbo:releaseDate(x, y) ← *dbo:Album(x); dbo:releaseDate(x, y)*
$M_{\mathbf{G}}$: is omitted here due to space limitation

and μ is the match predicate defined as

$$(s_1, s_2, s_3, o_1, \mathrm{o}_2, o_3) \in \mu \text{ iff } \sigma(s_k, o_k) \geq a, \text{ for each } k = 1, 2, 3$$

where σ is the 3-gram distance and α = 0.5. The match predicate compares the *title*, *artistName* and *releaseDate* of instances of *Record* from both views **F** and **G**.

3 Linkset Incremental Maintenance Based on Changesets

In this section, we present our approach to correctly compute the changeset for a linkset view **L**, based on the changesets published by $S_{\mathbf{F}}$ and $S_{\mathbf{G}}$. A *changeset* of an RDF dataset $\boldsymbol{S}$ from the state $\sigma_{\boldsymbol{S}}(t_0)$ in time t_0 to the state $\sigma_{\boldsymbol{S}}(t_1)$ in time t_1 is a pair $<\Delta_S^-(t_0, t_1), \Delta_S^+(t_0, t_1)>$, where $\Delta^-{}_{\boldsymbol{S}}(t_0, t_1)$ is the set of triples removed from $\sigma_{\boldsymbol{S}}(t_0)$ and $\Delta_S^+(t_0, t_1)$ is the set of triples added $\sigma_{\boldsymbol{S}}(t_0)$ to create $\sigma_{\boldsymbol{S}}(t_1)$ (for a formal definition see [8]). The approach we suggest to compute $\Delta_{\mathbf{L}}(t_0,t_1)$ follows two main steps: (1) Compute $R_{\mathbf{F}}(t_0,t_1)$, the set of resources that are affected by $\Delta_{\mathbf{SF}}(t_0,t_1)$ w.r.t **F**, and $R_{\mathbf{G}}(t_0,t_1)$, the set of resources that are affected by $\Delta_{\mathbf{SG}}(t_0,t_1)$ w.r.t **G**; (2) Compute $\Delta_{\mathbf{L}}(t_0,t_1)$ using the resources in $R_{\mathbf{F}}(t_0, t_1)$ and $R_{\mathbf{G}}(t_0, t_1)$.

3.1 Computing the Affected Resources

In this section we present an algorithm to compute $R_F(t_0,t_1)$, the set of resources that are affected by $\Delta_{\boldsymbol{SF}}(t_0,t_1)$ w.r.t **F**. We say that a resource $\boldsymbol{s}$ is *affected* by $\Delta_{\mathbf{SF}}(t_0,t_1)$ iff the state of $\boldsymbol{s}$ in $\sigma_{\mathbf{F}}(t_o)$ is different from the state of $\boldsymbol{s}$ in $\sigma_{\mathbf{F}}(t_1)$. More formally, a resource $\boldsymbol{s}$ *is affected by* $\Delta_{\boldsymbol{SF}}(t_0,t_1)$ w.r.t **F** iff $\boldsymbol{s}[\sigma_{\boldsymbol{F}}(t_o)] \neq \boldsymbol{s}[\sigma_{\mathbf{F}}(t_1)]$. To compute $R_{\mathrm{F}}(t_0,t_1)$, we have to consider two situations:

(i) If all mappings in M_F are simple mappings, $R_F(t_0,t_1)$ can be directly computed from $\Delta_{\mathbf{SF}}(t_0,t_1)$ [8].
(ii) Otherwise, the computation of $R_{\mathrm{F}}(t_0,t_1)$ requires, besides $\Delta_{\boldsymbol{SF}}(t_0,t_1)$, the old state $\sigma_{\mathbf{SF}}(t_0)$ of $S_{\mathbf{F}}$. But, $\sigma_{\mathbf{SF}}(t_0)$ is no longer available when the changeset is published.

To account for the second case, we introduce the notion of *auxiliary view* $\mathbf{A_F}$ for **F**, defined as a triple $\mathbf{A_F} = (V_{\mathbf{AF}}, S_{\mathbf{AF}}, M_{\mathbf{AF}})$, where:

- $V_{\mathbf{AF}}$ consists of all classes and properties in $V_{\mathbf{F}}$ that are relevant to S_{F}
- $S_{\mathbf{AF}} = S_{\mathbf{F}}$
- $M_{\mathbf{AF}}$ is a set of direct mappings from the vocabulary of $S_{\mathbf{F}}$ to $V_{\mathbf{AF}}$.

In the suggested architecture (see [8]), the auxiliary view $\mathbf{A_F}$ is materialized, while the view **F** is virtual. Algorithm 1, shown in Table 1, computes $R_F(t_0,t_1)$ when an auxiliary view is required. Theorem 1 in [8] shows that Algorithm 1 correctly computes the set of affected resources.

Table 1. Algorithm 1

```
Input: $\sigma_{\mathbf{AF}}(t_0)$, $\Delta_{\mathbf{SF}}(t_0,t_1)$
Step 1.1: Compute $\Delta^{-}_{\mathbf{AF}}(t_0, t_1) = M_{\mathbf{AF}}[\Delta^{-}_{\mathbf{SF}}(t_0, t_1)]$ and $\Delta^{+}_{\mathbf{AF}}(t_0, t_1) = M_{\mathbf{AF}}[\Delta^{+}_{\mathbf{SF}}(t_0, t_1)]$;
Step 1.2: Compute $R^{-}(t_0, t_1) = \{s$ / $s$ is the subject of a triple $t$ in $\sigma_{\mathbf{F}}(t_0)$ and
                                   $t$ is affected by a triple in $\Delta^{-}_{\mathbf{AF}}(t_0,t_1)\}$;
Step 1.3: Compute $\sigma_{\mathbf{AF}}(t_1) = (\sigma_{\mathbf{AF}}(t_0) - \Delta^{-}_{\mathbf{AF}}(t_0, t_1)) \cup \Delta^{+}_{\mathbf{AF}}(t_0, t_1)$;
Step 1.4: Compute $R^{+}(t_0, t_1) = \{s$ / $s$ is the subject of a triple $t$ in $\sigma_{\mathbf{F}}(t_1)$ and
                                   $t$ is affected by a triple in $\Delta^{+}_{\mathbf{AF}}(t_0,t_1)\}$;
Step 1.5: Return $R_{\mathbf{F}}(t_0, t_1) = R^{-}(t_0, t_1) \cup R^{+}(t_0, t_1)$.
```

To illustrate the computation of $R_F(t_0,t_1)$ by Algorithm 1, consider the linkset view **L** over the catalogue views **F** and **G** exported from *DBpedia* and *MusicBrainz*, defined in Sect. 2.2. The auxiliary view for **F** is $\mathbf{A_F} = (V_{\mathbf{AF}}, S_{\mathbf{AF}}, M_{\mathbf{AF}})$, where: $V_{\mathbf{AF}}$ = {*dbo: Album, foaf:name, dbo:artist, dbo:releaseDate* }; S_{AF}: http://host/dbpedia; $M_{\mathbf{AF}}$ is a set of direct mappings from *DBpedia's* vocabulary to $V_{\mathbf{AF}}$. The auxiliary view for **G** is $\mathbf{A_G} = (V_{\mathbf{AG}}, S_{\mathbf{AG}}, M_{\mathbf{AG}})$, where: $V_{\mathbf{AG}}$ = {*mo:Record, dc:title, foaf:maker, foaf:name, mo: realese, dc:date*}; $S_{\mathbf{AF}}$: http://host/MusicBrainz; $M_{\mathbf{AG}}$ is a set of direct mappings from *MusicBrainz's* vocabulary to $V_{\mathbf{AG}}$. Assume that:

- Table 2 shows the states of the catalogue views **F** and **G**, the auxiliary view $\mathbf{A_F}$ and linkset view **L**, on Sep 10, 2015 at 10:00 AM (t_0) and the triples published by the DBpedia Live extractor for the changes made on Sep 10, 2015 between 10:00 AM (t_0) and 11:02 PM (t_1).
- *MusicBrainz* did not release new changeset on Sep 10, 2015 between 10:00 AM (t_0) and 11:02 PM (t_1).

Algorithm 1 computes the set $R_F(t_0,t_1)$ in 5 steps:

Step 1.1: Compute $\Delta^{-}_{\mathbf{AF}}(t_0,t_1)$ and $\Delta^{+}_{\mathbf{AF}}(t_0,t_1)$. From Algorithm 1, we have:
$\Delta^{-}_{\mathbf{AF}}(t_0,t_1)$ = {(dbr:b1 *foaf:name* “Jackson Michael“)}.
$\Delta^{+}_{\mathbf{AF}}(t_0,t_1)$ ={(dbr:a1 *dbo:releaseDate* “1982-11-29”), (dbr:a1 *foaf:name* “Thriller”), (dbr:b1 *foaf:name* “Michael Joseph Jackson“)}.

Table 2. $\boldsymbol{\sigma}_{\mathbf{F}}(t_0)$, $\boldsymbol{\sigma}_{\mathbf{G}}(t_0)$, $\boldsymbol{\sigma}_{\mathbf{AF}}(t_0)$, $\boldsymbol{\sigma}_{\mathbf{AF}}(t_1)$, $\boldsymbol{\sigma}_{\mathbf{L}}(t_0)$, $\Delta^{-}_{DBpedia}(t_0, t_1)$ and $\Delta^{+}_{DBpedia}(t_0, t_1)$

$\sigma_F(t_0)$ = {
(dbr:a1 rdf:type *mo:Record*);
(dbr:a1 *mvl:artistName* “Jackson Michael“);
(dbr:a2 rdf:type *mo:Record*);
(dbr:a2 *dc:title* “Thriller 25”);
(dbr:a2 *mvl:artistName* “Jackson Michael“);
(dbr:a2 *dbo:releaseDate* “2008-02-08”)}
$\sigma_{AF}(t_0)$ = {
(dbr:a1 rdf:type *dbo:Album*);
(dbr:a1 *dbo:artist* dbr:b1);
(dbr:a2 rdf:type *dbo:Album*);
(dbr:a2 *foaf:name* “Thriller 25”);
(dbr:a2 *dbo:releaseDate* “2008-02-08”);
(dbr:a2 *dbo:artist* dbr:b1);
(dbr:b1 *foaf:name* “Jackson Michael“)}
$\sigma_{AF}(t_1)$ = {
(dbr:a1 rdf:type *dbo:Album*),
(dbr:a1 *foaf:name* “Thriller”),
(dbr:a1 *dbo:releaseDate* “1982-11-29”),
(dbr:a2 rdf:type *dbo:Album*),
(dbr:a2 *foaf:name* “Thriller 25”),
(dbr:a2 *dbo:releaseDate* “2008-02-08”),
(dbr:a1 *dbo:artist* dbr:b1),
(dbr:a2 *dbo:artist* dbr:b1),
(dbr:b1 *foaf:name* “Michael Joseph Jackson“)}
$\sigma_G(t_0)$ = {
(mbr:r1 rdf:type *mo:Record*);
(mbr:r1 *dc:title* “Thriller”);
(mbr:r1 *mvl:artistName* “Michael Joseph Jackson”);
(mbr:r1 *dc:releaseDate* “1982-11-29”);
(mbr:r2 rdf:type *mo:Record*);
(mbr:r2 *dc:title* “Thriller 25”);
(mbr:r2 *mvl:artistName* “Michael Joseph Jackson”);
(mbr:r2 *dc:releaseDate* “2008-02-08”)}
$\sigma_L(t_0)$ = {(dbr:a2 *owl:sameAs* mbr:r2)}
$\Delta^{-}_{DBpedia}(t_0, t_1)$ = {
(dbr:b1 *foaf:name* “Jackson Michael“)}
$\Delta^{+}_{DBpedia}(t_0, t_1)$ = {
(dbr:a1 *dbo:releaseDate* “1982-11-29”);
(dbr:a1 *foaf:name* “Thriller”);
(dbr:b1 *foaf:name* “Michael Joseph Jackson“)}

Step 1.2: Compute $R^{-}(t_0,t_1)$. First we have to compute which triples in $\sigma_F(t_0)$ are affected by triples in $\Delta^{-}_{\mathbf{AF}}(t_0,t_1)$. For example, consider the triple $\boldsymbol{y}$ = (dbr:b1 *foaf:name* “Jackson Michael”) in $\Delta^{-}_{\mathbf{AF}}(t_0, t_1)$. The triples (dbr:a1 *mvl:artistName* “Jackson Michael“) and (dbr:a2 *mvl:artistName* “Jackson Michael”) in $\sigma_F(t_0)$ are affected by $\boldsymbol{y}$ because those triples are generated by substituting *foaf:name(z, y)* by $\boldsymbol{y}$ in the mapping rule *“mvl:artistName(x, y) ← dbo:Album(x); dbo:artist(x, z); foaf:name(z, y)”*. Therefore, $R^{-}(t_0,t_1)$ = {dbr:a1, dbr:a2}.
Step 1.3: Compute $\sigma_{\mathbf{AF}}(t_1) = (s_{\mathbf{AF}}(t_0) - \Delta^{-}_{\mathbf{AF}}(t_0, t_1)) \cup \Delta^{+}_{\mathbf{AF}}(t_0, t_1)$ (See Table 2).
Step 1.4: Compute $R^{+}(t_0, t_1)$. First we have to compute the triples in $\sigma_F(t_1)$ that are affected by triples in $\Delta^{+}_{\mathbf{AF}}(t_0, t_1)$. The triples (dbr:a1 *dbo:releaseDate* “1982-11-29”), (dbr:a1 *dc:title* “Thriller”), and (dbr:a1 *mvl:artistName* “Jackson Joseph Michael”) in $\sigma_F(t_1)$ are affected by triples in $\Delta^{+}_{\mathbf{AF}}(t_0, t_1)$. Therefore, $R^{-}(t_0, t_1)$ = {dbr:a1, dbr:a2}.
Step 1.5: Compute $R_{\mathbf{F}}(t_0,t_1) = R^{-}(t_0, t_1) \cup R^{+}(t_0, t_1)$
$R_{\mathbf{F}}(t_0,t_1)$ = {dbr:a1, dbr:a2}.

3.2 Computing the Changeset for L

Algorithm 2 in Table 3 returns $\Delta_{\mathbf{L}}(t_0,t_1)$, a changeset for **L**. Theorem 2 in [8] shows that the changeset $\Delta_{\mathbf{L}}(t_0,t_1)$ returned by Algorithm 2 correctly maintains **L**. To illustrate

Table 3. Algorithm 2

```
Input: $\sigma_L(t_0)$, $\sigma_{AF}(t_1)$, $\sigma_{AG}(t_1)$, $R_F(t_0,t_1)$, $R_G(t_0,t_1)$
Step 2.1: Compute
        $D_F = \{ (s, p, o) / (s, p, o) \in \sigma_L(t_0)$ and $s \in R_F(t_0,t_1)\}$
        $I_F = \{ (s, p, o) / s \in R_F(t_0,t_1), o \in M_F[\sigma_{AF}(t_1)]$;
Step 2.2: Compute
        $D_G = \{ (s, p, o) / (s, p, o) \in \sigma_L(t_0)$ and $s \in R_G(t_0,t_1)\}$
        $I_G = \{ (s, p, o) / s \in R_G(t_0,t_1), o \in M_G[\sigma_{AG}(t_1)]$;
Step 2.3: Compute $\Delta^{-}_{L}(t_0,t_1) = D_F \cup D_G$;
Step 2.4: Compute $\Delta^{+}_{L}(t_0,t_1) = I_F \cup I_G$;
Step 2.5: Return $\Delta_L(t_0, t_1) = < \Delta^{-}_{L}(t_0,t_1), \Delta^{+}_{L}(t_0,t_1) >$ .
```

the computation of $\Delta_L(t_0,t_1)$ by Algorithm 2, consider the set $R_F(t_0,t_1)$ computed in Sect. 3.1. Since we assume that *MusicBrainz* did not release new changesets in the time interval considered, $R_G(t_0,t_1)=\emptyset$. Algorithm 2 computes $\Delta_L(t_0,t_1)$ in 5 steps.

Step 2.1: Compute D_F and I_F. From Algorithm 2, we have:
D_F = {(dbr:a2 *owl:sameAs* mbr:r2)};
I_F = {(dbr:a1 *owl:sameAs* mbr:r1), (dbr:a2 *owl:sameAs* mbr:r2)}.
Step 2.2: Compute D_G and I_G. From Algorithm 2, we have:
$D_G = \emptyset$; $I_G = \emptyset$.
Step 2.3: Compute $\Delta^{-}_{L}(t_0, t_1)$. From Algorithm 2, we have:
$\Delta^{-}_{L}(t_0, t_1)$ = {(dbr:a2 *owl:sameAs* mbr:r2)}.
Step 2.4: Compute $\Delta^{+}_{L}(t_0, t_1)$. From Algorithm 2, we have:
$\Delta^{+}_{L}(t_0, t_1)$ ={(dbr:a1 *owl:sameAs* mbr:r1), (dbr:a2 *owl:sameAs* mbr:r2)}.
Step 2.5: Return $\Delta_L(t_0, t_1) = < \Delta^{-}_{L}(t_0, t_1), \Delta^{+}_{L}(t_0, t_1) >$

The new state of **L** is computed by $\sigma_L(t_1) = (\sigma_L(t_0) - \Delta^{-}_{L}(t_0, t_1)) \cup \Delta^{+}_{L}(t_0, t_1)$. Therefore, $\sigma_L(t_1)$ ={(dbr:a1 *owl:sameAs* mbr:r1), (dbr:a2 *owl:sameAs* mbr:r2)}.

4 Implementation and Experiments

The *Linkset Maintainer* tool was developed using Java, JBoss 7, Open Link Virtuoso as the triple store, and Silk as the link discovery tool. In order to evaluate the performance of the incremental strategy, we selected two datasets: a *Music Brainz* dump, and the *DBpedia* endpoint. Additionally, *DBpedia* daily provides sets of changed triples extracted from Wikipedia, called *DBpedia* Changesets (available at http://live.dbpedia.org/changesets/), which are organized by year, month, day, and hour; and also separated by the type of update (added, removed, reinserted, and clear).

We defined views about music records released after 2010 for each dataset. The “MusicBrainz_Records” view had 311,374 resources and the “DBpedia_Records” view had 35,651resources. Then, we materialized an *owl:sameAs* linkset of records, using these views, by comparing their titles, artist names and release dates. The linkset had 14,716 links and the runtime to compute it using Silk was around 4 h. To test the performance of the incremental strategy, we processed and analyzed one entire day (October 3th, 2015) of DBpedia changesets. We computed the total number of inserted and deleted resources, the sets $\Delta^{-}_{\mathbf{AF}}(t_0, t_1)$, $\Delta^{+}_{\mathbf{AF}}(t_0, t_1)$, R^{-} and R^{+}, and the runtime to maintain the linkset. Table 4 summarizes the results for the whole day.

Table 4. Analysis of DBpedia Changesets.

	Total	Sets	Avg	Max
Deleted resources	144385	720	200,5	1230
Auxiliary view deleted resources $\left(\Delta^{-}_{\mathbf{AF}}(t_0, t_1)\right)$	7453	720	10,35	85
View deleted resources (R^{-})	318	720	0,4	16
Inserted resources	134887	720	187,3	1072
Auxiliary view inserted resources $\left(\Delta^{+}_{\mathbf{AF}}(t_0, t_1)\right)$	7614	720	10,58	85
View inserted resources (R^{+})	372	720	0,5	16
Runtime	12 h	720	1 min	5 min

Note that, on the average, there are 200,5 deleted resources per changeset, of which only an average of 0,4 resources affected the view. Also note that the max number of R^{-} and R^{+} was only 85. We highlight that the max runtime to process a changeset was 5 min, which included the time to download and decompress the changeset file, compute R^{-} and R^{+}, and update the linkset. Recall that the runtime to materialize the linkset was 4 h, which is much higher than the max runtime to incrementally maintain the linkset using a changeset. Therefore, in these experiments, we showed that the incremental strategy, by far, outperformed the re-materialization strategy.

5 Conclusions

Data publishers frequently materialize linkset views between two source datasets using link discovery tools. However, when the source datasets are updated, the materialized linkset views must also be updated. To help solve this problem, we presented a formal framework for maintaining linkset views that adopts changesets and an incremental strategy. The changesets, published by the source datasets, are used to compute the set of updated resources that are relevant to a linkset view; the incremental strategy updates the links only for the relevant resources. We provided a formalization of our approach and indicated that the framework correctly maintains the linkset views. We also described experiments to validate the proposed framework.

Acknowledgments. This work was partly funded by CNPq under grants 153908/ 2015-7, 557128/2009-9, 444976/2014-0, 303332/2013-1, 442338/2014-7 and 248743/ 2013-9, by FAPERJ under grants e E-26-170028/2008 and E-26/201.337/2014 and by FCT – Fundação para a Ciência e Tecnologia, under grant SFRH/BPD/76024/2011.

References

1. Berners-Lee, T.: (2006). http://www.w3.org/ DesignIssues/LinkedData.html
2. Casanova, M.A., Vidal, V.M.P., Lopes, G.R., Leme, L.A.P.P., Ruback, L.: On materialized sameAs linksets. In: 25th International Conference on Database and Expert Systems Applications, pp. 377–384 (2014)
3. Endris, M.K., Faial, S., Orlandi, F., Auer, S., Scerri, S.: Interest-based RDF update propagation. In: 14th International Semantic Web Conference, pp. 650–665 (2015)
4. Isele, R., Jentzsch, A., Bizer, C.: efficient multidimensional blocking for link discovery without losing recall. In: 14th International Workshop on the Web and Databases (2011)
5. Ngomo, A.C.N., Auer, S.: Limes: a time-efficient approach for large-scale link discovery on the web of data. In: 22nd International Joint Conference on Artificial Intelligence, pp. 2312–2317 (2011)
6. Popitsch, N., Haslhofer, B.: DSNotify – a solution for event detection and link maintenance in dynamic triplesets. J. Web Semant. **9**(3), 266–283 (2011)
7. Vidal, V.M.P., et al.: Specification and incremental maintenance of linked data mashup views. In: 27th International Conference on Advanced Information Systems Engineering, pp. 214–229 (2015)
8. Vidal, V.M.P., Casanova, M.A., Menendez, E.S., Arruda, N.: A formal approach based on changesets for the incremental maintenance of linkset views. Technical report 015, Department of Informatics, PUC-Rio, Brazil (2015)
9. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Discovering and maintaining links on the web of data. In: 8th International Semantic Web Conference, pp. 650–665 (2009)

Spatial and Temporal Data

Graph-Based Metric Embedding for Next POI Recommendation

Min Xie[1], Hongzhi Yin[2](✉), Fanjiang Xu[1], Hao Wang[1], and Xiaofang Zhou[2]

[1] Science and Technology on Integrated Information System Laboratory,
Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
{xiemin2014,fanjiang,wanghao}@iscas.ac.cn

[2] School of Information Technology and Electrical Engineering,
The University of Queensland, Brisbane, QLD 4072, Australia
db.hongzhi@gmail.com, zxf@itee.uq.edu.au

Abstract. With the rapid prevalence of smart mobile devices and the dramatic proliferation of location-based social networks (LBSNs), point of interest (POI) recommendation has become an important means to help people discover attractive and interesting places. In this paper, we investigate the problem of next POI recommendation by considering the sequential influences of POIs, as a natural extension of the general POI recommendation, but it is more challenging than the general POI recommendation, due to that (1) users' preferences are dynamic, and the next POI recommendation requires tracking the change of user preferences in a real-time manner; and (2) the prediction space is extremely large, with millions of distinct POIs as the next prediction target, which impedes the application of classical Markov chain models. In light of the above challenges, we propose a graph-based metric embedding model which converts POIs in a low dimensional metric and tracks the dynamics of user preferences in an efficient way. Besides, the knowledge of sequential patterns of users' check-in behaviors can be exploited and encoded in the POI embedding, which avoid the time-consuming computation of the POI-POI transition matrix or even cube as the Markov chain-based recommender models have done. In other words, our proposed method effectively unifies dynamic user preferences and sequential influence via the POI embedding. Experiments on two real large-scale datasets demonstrate a significant improvement of our proposed models in terms of recommendation accuracy, compared with the state-of-the-art methods.

Keywords: Next POI recommendation · Metric embedding · Sequential influence · Dynamic user preferences

1 Introduction

With the rapid development of Web 2.0, location acquisition and wireless communication technologies, a sufficient number of location-based social networks (LBSNs) have emerged in recent years, such as Foursquare, Facebook Places,

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 207–222, 2016.
DOI: 10.1007/978-3-319-48743-4_17

Gowalla and Loopt, where users can check in at point-of-interests (POIs), e.g., stores, restaurants, sightseeing sites, and share their life experiences in the physical world via mobile devices. To help users navigate a huge number of POIs and suggest the most suitable POIs to meet their personal preferences, POI recommendation has become an important means and played a critical role in LBSN services. POI recommendation aims at learning users' preferences based on their check-in records and then predicting users' preferred POIs for recommendation. Recently, many various recommender models have been proposed for POI recommendation by exploiting and integrating geographical influence [18], social influence [5], temporal cyclic effect [6,26], word-of-mouth effect [9,25], content effect [15,24] and their joint effect [9,22,25].

Next POI recommendation [2], as a natural extension of general POI recommendation, is recently proposed. There are relatively few studies on this new problem and it is very challenging. Different from general POI recommendation that focuses on estimating users' static preferences on POIs, next POI recommendation requires provides satisfactory recommendations promptly based on users' latest preferences and their most recent checked-in POIs, which requires producing recommendation results in a real-time manner. However, most of existing general POI recommender models is incapable of supporting real-time recommendation, and they would suffer from the following two drawbacks: (1) Delay on model updates caused by the expensive time cost of re-running the recommender model; and (2) Disability to track changing user preferences due to the fact that latest check-in records used for updating recommendation models are often overwhelmed by the large data of the past. Accurately capturing the change of user preferences in a real-time manner is very helpful for next POI recommendation. Since each check-in provides valuable information about the user's preferences, recommender model must respond immediately to new check-in information.

On the other hand, several Markov chain-based recommender models [2,28,29] have been recently developed to capture the sequential patterns of POIs. But, they encounter from the huge parameter prediction space. Suppose there are a collection of V POIs and the next POI depends on the previous n ones. These recommendation methods then need to estimate $|V|^{n+1}$ free parameters in the nth order Markov chain model, which is extremely computational-expensive. To reduce the size of the prediction space, most related studies [2,29] exploit sequential influence using a first-order Markov chain, which considers only the last one in a sequence of locations visited by a user to recommend a new location for her. Although the parameter space can be decreased to $|V|^2$, it may still be huge considering that V is usually a large number in LBSNs. Hence, we aim to develop a new method with a small number parameters to incorporate the influence from all recently visited locations, rather than just the last one.

More recently, methods of embedding items in a low-dimension Euclidean space have been widely adopted in a variety of fields, including natural language processing, text mining and music information retrieval. Tang et al. [13] predicted text embeddings based on heterogeneous text networks which showed great potential in document classification. Chen et al. [1] proposed a Logistic

Markov embedding (LME) to map each song to one point (or multiple points) in a latent Euclidean space for playlists generating, which also verifies the effectiveness of embedding methods.

In this paper, we stand on the recent advances in embedding learning techniques and propose a graph-based metric embedding method called GME to effectively learn the embeddings of POIs in a low-dimension Euclidean space. Then, we track the dynamic user preferences and provide recommendations based on the embeddings of the user's check-in POIs and their timestamps. Specifically, we adopt a time-decay manner to compute the user's dynamic preferences from his/her checked-in POIs, i.e., if a POI is visited by the user more recently, it will be more important and assigned with a higher weight. Just like the classic item-based collaborative filtering method [8], our proposed recommendation method has the nice properties of making fast response to new check-in information, producing dynamic recommendations in realtime and scaling to massive data sets, and our GME model only needs to be trained once to obtain the embeddings of POIs. Note that the computation of POI-POI correlation (or similarity) in our method is based on the learnt POI embeddings, which effectively overcomes the issue of data sparsity encountered by the item-based CF. To further improve the effectiveness of our method in next POI recommendation, we extend the GME model to GME-S by exploiting the sequential patterns of POIs. Since the knowledge of sequential patterns is encoded in the POI embeddings, we do not need to integrate the sequential influence to next POI recommendation in an explicit way as the Markov chain-based recommendation method do. In our GME-S model, the parameter space is $|V| \times d$ where d is the dimension of POI embedding that tends to be smaller than 100. Thus, the parameter space in our model is much smaller than $|V| \times |V|$.

To summarize, we make the following contributions:

- We develop a graph-based metric embedding (GME) model to learn the representation of POIs in a low-dimension latent space. Then, we propose a time-decay method to track and represent the dynamic user preferences based on the learnt POI embeddings.
- To model the sequential influence of POIs, we further extend our GME to GME-S model by exploiting and integrating the sequential patterns in the learning process of POI embeddings. To the best of our knowledge, this is the first work that uses the metric embedding method to unify dynamic user preferences and the sequential influence in a principled manner.
- We conduct comprehensive experiments to evaluate the performance of our proposed methods on two large scale real datasets. The results show the superiority of our proposals in recommending next POIs for users by comparing with the state-of-the-art techniques.

The remainder of the paper is organized as follows. Section 2 details our proposed recommendation approach. We report the experimental results in Sect. 3. Section 4 reviews the related work and we conclude the paper in Sect. 5.

2 Graph-Based Metric Embedding

In this section, we first formulate the problem definitions, and then present our proposed Graph-based Metric Embedding (GME) model, as well as its extension GME-S which incorporate the sequential influence.

2.1 Problem Definitions

For ease of presentation, we define the key data structures and notations used in this paper. Table 1 also lists them.

Table 1. Notations used in this paper.

Variable	Interpretation
U, V	The set of users and POIs
D_u	The profile of user u
$\mathbb{R}^d$	d dimensional metric
$\boldsymbol{p}_{u,t}, \boldsymbol{q}_v$	Time-aware user embedding and POI embedding
S_u	The sequence of user u
$\triangle T$	The time period threshold

Definition 1 ***(POI).*** *A POI is defined as a uniquely identified specific site (e.g., a restaurant or a cinema). We use v to represent a POI.*

Definition 2 ***(Check-in Activity).*** *A user check-in activity is represented by a triple (u, v, t) that means user u visits POI v at time t.*

Definition 3 ***(User Profile).*** *For each user u, we create a user profile D_u, which is a set of check-in activities associated with u. The dataset D used in our model includes all user profiles, i.e., $D = \{D_u : u \in U\}$.*

Since we use graph-based method to embed POIs, now let us begin with formally defining of the POI-POI graph, POI embedding and user embedding. The toy example of generating the POI-POI graph is show in Fig. 1.

Definition 4 ***(POI-POI Graph).*** *A POI-POI co-occurrence graph, denoted as $G = (V, E)$, captures the POI co-occurrence information in a user profile D_u. V is a set of POIs and E is the set of edges between POIs. In general, if a user u has checked in two POIs v_i and v_j, there will be an edge e_{ij} between v_i and v_j. The weight w_{ij} of edge e_{ij} is defined as the number of times that the two POIs co-occured in the whole dataset D.*

Definition 5 ***(POI Embedding).*** *Each POI v in the dataset D will be represented by a POI embedding $\boldsymbol{q}_v$ in the $\mathbb{R}^d$ metric.*

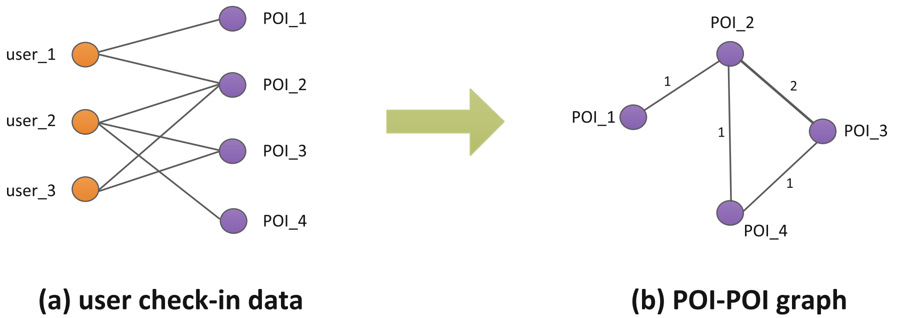


Fig. 1. A toy example of generating POI-POI graph from user profile D_u.

Definition 6 ***(User Embedding).*** *For each user u in the dataset D, his/her dynamic preferences will be represent as a time-aware user embedding $\boldsymbol{p}_{u,t}$ in the $\mathbb{R}^d$ metric.*

The POI-POI graph captures the POI co-occurrences which is resemble to the item-based collaborative filtering that mines the item similarity. Our goal is to project POIs in a low dimensional metric $\mathbb{R}^d$ through the graph information, thus we can get the POI embedding $\boldsymbol{q}$, then we calculate the cosine similarity between POIs. Intuitively, if two POIs are often checked in together, their cosine similarity will be larger in $\mathbb{R}^d$: if a user has visited one of them, he is probably to check in the other one.

Given a dataset D as the union of a collection of user profiles, we are aim to provide next POI recommendations for querying users and time, stated as follows.

Problem 1 **(Next POI Recommendation).** Given a user activity dataset D and a querying user u at time t (i.e., the query is $q = (u, t)$), our goal is to recommend a list of POIs that u would be interested in next.

2.2 Model Description and Inference

In this section, we first propose a graph-based embedding model (GME) to learn POI representation in the latent space, and then present how to track and represent the dynamic user preferences.

Predictive POI Embedding. Given the POI-POI graph $G = (V, E)$, where V is the set of POIs and E is the set of edges between them. For each edge e_{ij} whose source node is v_i, target node is v_j in the graph, we first define the conditional probability of vertex v_j generated vertex v_i as:

$$p(v_j|v_i) = \frac{\exp(\boldsymbol{q}_j^{\mathrm{T}} \cdot \boldsymbol{q}_i)}{\sum_{k=1}^{|V|} \exp(\boldsymbol{q}_k^{\mathrm{T}} \cdot \boldsymbol{q}_i)} \tag{1}$$

where $\boldsymbol{q}_i$ is the embedding vector of vertex v_i, and $\boldsymbol{q}_j$ is the embedding vector of vertex v_j, Eq. 1 defines a conditional distribution $p(\cdot|v_i)$ over all the vertices. To preserve the weight w_{ij} on edge e_{ij}, we can make the conditional distribution $p(\cdot|v_i)$ be close to its empirical distribution $\hat{p}(\cdot|v_i)$, which can be defined as $\hat{p}(v_j|v_i) = \frac{w_{ij}}{deg_i}$. Then minimize the following objective function:

$$O = \sum_{i \in V} \lambda_i d(\hat{p}(\cdot|v_i), p(\cdot|v_i)) \tag{2}$$

where $d(\cdot, \cdot)$ is the KL-divergence between two distributions, λ_i is the importance of vertex v_i in the network, which can be set as the degree $deg_i = \sum_j w_{ij}$. Omitting some constants, the objective function Eq. 2 can be calculated as:

$$O = -\sum_{(i,j) \in E} w_{ij} \log p(v_j|v_i) \tag{3}$$

By learning $\{\boldsymbol{q}_i\}_{i=1...|V|}$ that minimize Eq. 3, we are able to represent every POI v_i with a d dimensional embedding $\boldsymbol{q}_i$ in metric $\mathbb{R}^d$.

Model Inference. Optimizing objective function Eq. 3 is computationally expensive, as calculating the conditional probability $p(\cdot|v_i)$ need to sum over the entire set of vertices. To address this problem, we sample multiple negative edges according to some noisy distribution for each edge e_{ij} following the negative sampling approach proposed in [10]. For each edge e_{ij}, it specifies the following objective function:

$$\log \sigma(\boldsymbol{q}_j^{\mathrm{T}} \cdot \boldsymbol{q}_i) + \sum_{n=1}^{K} E_{v_n \sim P_n(v)}[\log \sigma(-\boldsymbol{q}_n^{\mathrm{T}} \cdot \boldsymbol{q}_i)] \tag{4}$$

where $\sigma(x) = 1/(1 + exp(-x))$ is the sigmoid function, K is the number of negative edges. We set $K = 5$, $P_n(v) \propto d_v^{3/4}$ from the empirical setting of [10], d_v is the out-degree of vertex v. Then we adopt the asynchronous stochastic gradient algorithm (ASGD) [11] for Eq. 4. If an edge e_{ij} is sampled, the gradient w.r.t. the embedding vector $\boldsymbol{q}_i$ of vertex v_i will be calculated as:

$$\frac{\partial O}{\partial \boldsymbol{q}_i} = w_{ij} \cdot \frac{\partial \log p(v_j|v_i)}{\partial \boldsymbol{q_i}} \tag{5}$$

However, when the weights of edges have a high variance there will be a problem, because it is very hard to find a good learning rate. If we select a large learning rate according to the edges with small weights, the gradients on edges with large weights will explode, while the gradients will become too small if we select the learning rate according to the edges with large weights. To overcome this dilemma, we follow the edge sampling approach using in [14]. Let $W = (w_1, w_2, ..., w_{|E|})$ denote the ranking sequence of edge weights. First, we calculate the sum of the weights $w_{sum} = \sum_{i=1}^{|E|} w_i$. Then, sample a value within $[0, w_{sum}]$ to see which interval $[\sum_{j=0}^{i-1} w_j, \sum_{j=0}^{i} w_j)$ the random value falls into.

In the latter procedure, we use alias table method [7] to draw a sample, thus reduce the sampling complexity to $O(1)$. Moreover, optimization with negative sampling takes $O(\eta \times (K+1))$ time cost, where K is the number of negative samples and η is the time taking for one sampling. Thus, the entire step takes $O(\eta \times K)$ time. In fact, the number of steps used for optimization is usually proportional to the number of edges $|E|$. Therefore, the overall time complexity of optimization is $O(\eta \times K \times |E|)$, while η, K are all constants. The proposed edge sampling method is very efficient since it is linear to the number of edges $|E|$, and does not depend on the number of vertices $|V|$.

Predictive Dynamic User Embedding. General recommender models (e.g. latent factor models) achieves the dynamic update of user preferences via re-training the model or applying the online learning techniques, which is very time-consuming. We aim to propose an efficient approach that tracks the dynamic of user preferences in a linear time complexity. To achieve this, we map dynamic user preferences to the same dimensional metric $\mathbb{R}^d$ as POIs, and utilize the learnt POI embeddings to represent the dynamic user preference embedding. More precisely, we assume that an individual's preferences at time t are affected by the whole set of POIs he has visited in the user profile D_u before time t. Note that, the check-ins in D_u are ranked according to their check-in timestamps in an increasing order. Therefore, we can learn the embedding $\boldsymbol{p}_{u,t}$ of u's preferences at time t by utilizing the vectors of POIs he has visited before t in the form of exponential decay. That is, if a user u has checked in a set of POIs before time t, his/her preferences at time t can be computed as:

$$\boldsymbol{p}_{u,t} = \sum_{(v_i,t_i)\in D_u \cap (t_i<t)} exp^{-(t-t_i)} \cdot \boldsymbol{q}_i \tag{6}$$

where $\boldsymbol{q}_i$ is the embedding of POI v_i, (v_i, t_i) is u's check-in record in D_u before time t; the later the POI is visited, the bigger the exponential is. In this way, we can dynamically track the user's preferences in a linear time.

2.3 Incorporating Sequential Influence

Since it has been shown in multiple studies that human movement in LBSNs clearly demonstrates sequential patterns [27,28], we further extend our GME model by incorporating sequential influence and propose GME-S model, which unifies the sequential influence and dynamic user preferences in a principled way. Intuitively, if two POIs have been checked in by a user together with a big time interval, they may not have a strong link. To describe our GEM-S model clearly, we first define the notations behind, and it is also listed in Table 1.

Definition 7 ***(Sequence).*** *A sequence of user u, consists of an ordered list of elements, denoted by $S_u = \{(v_1,t_1),(v_2,t_2),...,(v_n,t_n)\}$, where each element (v_i,t_i) indicates that user u visited POI v_i at time t_i ($1 \leq i \leq n$ and $t_1 \leq t_2 \leq ... \leq t_n$).*

In the GME-S model, there is also a kind of POI-POI graph, but it is different from that in the GME model as it incorporates the sequential patterns of POIs. To distinguish the two graphs, we call it sequential POI-POI graph which is defined as follow.

Definition 8 ***(Sequential POI-POI Graph).*** *A sequential POI-POI co-occurrence Graph, denoted as $G = (V, E)$, captures the check-in sequence of POIs in a user profile D_u. V is a set of POIs and E is the set of edges between POIs.* ***Given a time period threshold*** $\triangle T$, ***for each check-in pair*** $\{(v_i, t_i), (v_j, t_j)\}$ ***in user's sequence*** S_u, *if* $0 < t_i - t_j \leq \triangle T$,***there will be an edge*** e_{ij} ***between*** v_i ***and*** v_j. *The weight w_{ij} of edge e_{ij} is defined as the number of times that the two POIs sequentially co-occur in the whole dataset D.*

The learning algorithm for POI embeddings on the sequential POI-POI graph is the same as that of GME model, and we will study the impact of $\triangle T$ on the quality of next POI recommendation in Sect. 3.3. Thus, the sequential information is encoded in the POI embeddings.

2.4 Next POI Recommendation

Our proposed GME and GME-S models are employed to make next POI recommendation as follows. Given a user u at time t (that is, the query is $q = (u, t)$), our task is to recommend top-k POIs that u wishes to visit from the POIs that the user has not visited before. More precisely, given the user u and time t, for each POI v which has not been visited by u, we compute its ranking score as in Eq. 7, and then select the k ones with the highest ranking scores as recommendations.

$$S(u, v, t) = \boldsymbol{p}_{u,t}^{\mathrm{T}} \cdot \boldsymbol{q}_v \tag{7}$$

where $\boldsymbol{p}_{u,t}$ is the representation of u's preferences at time t, which can be computed in Eq. 6, and $\boldsymbol{q}_v$ is the embedding of POI v. From the above Equation, we can see that we do not explicitly integrate the sequential influence, as the sequential information has been captured by the POI embeddings. Thus, we avoid computing the huge POI-POI transition matrix or even cube as other Markov chain-based recommender model have done.

3 Experiments

In this section, we move forward to evaluate the effectiveness of the proposed GME and GME-S model for next POI recommendation. The experiments are set up as the following.

3.1 Experimental Settings

Datasets. Our experiments are performed on two real large-scale LBSNs datasets: Foursquare and Twitter. The basic statistics of them are shown in Table 2. The two real datasets are publicly available[1].

[1] https://sites.google.com/site/dbhongzhi/.

Table 2. Basic statistics of datasets

	Foursquare	Twitter
# of users	4,163	114,508
# of POIs	121,142	62,462
# of check-ins	483,813	1,434,668

Foursquare. This dataset contains 483,813 check-in histories of 4,163 users who live in the California, USA. The whole dataset covers 121,142 POIs around the world.

Twitter. This dataset is based on the publicly available Twitter dataset in [3]. The dataset contains 1,434,668 check-in histories of 114,508 users over 62,462 POIs.

Comparative Approaches. We compare our GME and GME-S with the following three methods representing the state-of-the-art next POI recommendation techniques.

BPR. BPR [12] is a generic Bayesian method for learning models for personalized ranking from implicit feedback which absolutely meet the top-k recommend requirements according to user check-in data compared to matrix factorization based methods. However, BPR does not contain sequential influence, thus we use it to compare with our proposed GME model.

SPORE. SPORE [16] is a sequential personalized POI recommendation framework, which introduces a novel latent variable topic-region to model and fuse sequential influences and personal interests in a latent and exponential space, which considers the user preferences and sequential influence simultaneously as the GME-S do.

PRME. PRME [4] is a personalized ranking metric embedding algorithm that jointly models the sequential transition of POIs and user preferences. It also exploits the metric embedding method for the next POI recommendation, but utilize two latent spaces: one is the sequential transition space and the other is the user preferences space.

Evaluation Methods. Given a user profile D_u in terms of a collection of user activities, we first extracted the activity sequence of each user S_u, then divide the user's activities into a train set and a test set, and make sure the timestamp of check-ins in the test set happened behind that in the train set. Besides this constraint, we randomly select 20 % of the activity records as test set and the rest the train set. Therefore, we split the user activity dataset D into the train set D_{train} and the test set D_{test}. To evaluate the recommendation methods, we adopt the evaluation methodology and measurement Accuracy@k proposed in

[5,19,21,24,25]. Specifically, for each activity record (u, v, t) in D_{test} as well as its associated query q we make the following procedure:

- First, to track user's current interests, we compute $\boldsymbol{p}_{u,t}$ which means the querying user u's preferences at time t on basis of Eq. 6.
- Second, we calculate the ranking score for POI v and all other POIs which are unvisited previously by u through Eq. 7.
- Third, according to the ranking scores of all these POIs, we form a ranked list ordered by the scores. Let p denote the position of v within this list. Obviously, we expect POI v precedes all the other unvisited POIs, which means $p = 1$.
- Fourth, we formed a top-k recommendation list by picking the k top ranked POIs from the list. If $p \leq k$ (i.e., the ground truth POI v appears in the top-k recommendation list), we have a hit. Otherwise, we have a miss.

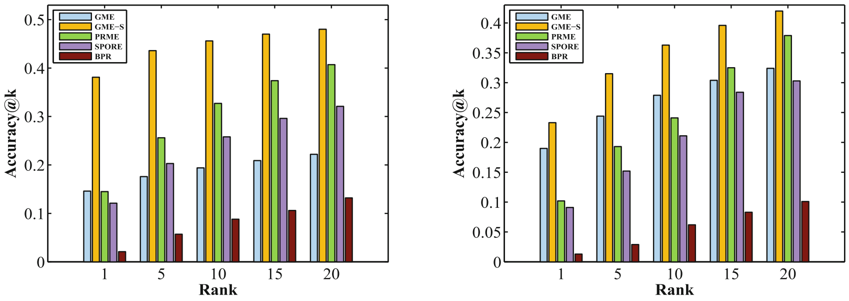
We define hit@k for a single test case as either the value 1, if the ground truth POI v appears in the top-k results, or the value 0, if otherwise. The overall Accuracy@k is defined by averaging over all test cases which proceeds as Eq. 8 shows:

$$Accuracy@k = \frac{\#hit@k}{|D_{test}|} \tag{8}$$

where #hit@k denotes the number of hits in the whole test set, and $|D_{test}|$ is the number of test cases.

3.2 Recommendation Effectiveness

In this part, we present the effectiveness of all the recommendation methods with well-tuned parameters. Figure 2 reports the performance of the recommendation methods on Foursquare and Twitter datasets respectively. Note that, we only show the performance when $k = \{1, 5, 10, 15, 20\}$, since a greater value of k is usually ignored for the top-k recommendation task.



(a) Performance on Foursquare

(b) Performance on Twitter

Fig. 2. Recommendation effectiveness

In our proposed methods, we set $d = 60$, $\triangle T = 5$ days on Foursquare dataset, while $d = 70$, $\triangle T = 20$ on Twitter dataset, the selection of these two parameters will be shown in Sect. 3.3. It is obvious that our proposed GME-S model outperforms other competitor models significantly, and GME also show a fairish result compared to the model without considering sequential influence. Several observations made from the results are presented following:

(1) GME and GME-S have a Higher Ability to Place Ground Truth in Top-1 Position. It is apparent that the competitor recommendation methods have significant performance disparity in terms of the top-k accuracy, while GME and GME-S will get a relatively high accuracy no matter how small the k is. Top-1 result takes up 65.8 % of top-20 result in GME and 79.4 % of top-20 result in GME-S on Foursquare (35.6 % in PRME, 37.7 % in SPORE and 15.9 % in BPR respectively). The result on Twitter dataset also verifies this phenomenon, top-1 result occupies 58.6 % of top-20 result in GME and 55.5 % of top-20 result in GME-S (26.9 % in PRME, 30.0 % in SPORE and 12.9 % in BPR).

(2) Sequential Influence Plays an Important role in next POI Recommendation. GME-S model is more efficient than GME model on both datasets which demonstrate the beneficial brought by incorporating sequential influence. The accuracy gap between GME and GME-S is bigger on Foursquare dataset than that on Twitter dataset is mainly because that the users in Foursquare dataset visited more POIs per capita than that in Twitter dataset, which may bring more noise edges in POI-POI graph in GME model. Moreover, PRME and SPORE outperform BPR and GME on Foursqaure dataset also shows the importance of sequential influence as these methods also considered the sequential factor.

(3) Metric Embedding Methods Outperform Other Competitors. GME, GME-S and PRME are all metric embedding based methods, they get a better efficiency than other competitors (GME-S and PRME outperform SPORE and GME outperforms BPR) which may imply consistent interpretation is important in top-k recommendation. Moreover, GME-S does better than PRME which may be because GME-S embeds all information into one latent space while PRME uses two latent spaces to embed user preferences and sequential patterns, respectively.

(4) Observations About the Datasets. GME performs better on the Twitter dataset than the Foursquare dataset, it is because the data covers 4 years on Foursquare dataset while half year on Twitter dataset, which brings in too many noise edges in POI-POI graph as we generate edges only consider POI co-occurrences in GME. However, the GME-S method can overcome this deficiency easily and get a high effectiveness. Meanwhile, the high user check-in density makes accuracy on Foursquare dataset is better than that on Twitter dataset in all other methods.

3.3 Impact of Model Parameters

Tuning model parameters is critical to the performance of the proposed models. There is the metric dimension d to be studied in GME and the metric dimension d and time period threshold $\triangle T$ to be tested in GME-S since we consider sequential influence in GME-S. The impact of d in GME is similarly to that in GME-S, we therefore show the impact of d and $\triangle T$ in GME-S in this subsection.

Impact of Metric Dimension $\boldsymbol{d}$**.** Table 3 depicts the impact of the metric dimension d. From the results, we observe that the recommendation accuracy of GME-S first increased with the increasing number of dimension d, and then it does not change significantly when the number of dimension is larger than 60 on Foursquare dataset and 70 on Twitter dataset. The reason is that high dimensions can better embody the latent metric relationships, but when d exceeds a threshold (e.g., $d = 60$ on Foursquare dataset and $d = 70$ on Twitter dataset), the dimension is enough to embed the relationships. At this point, it is less helpful to improve the model performance by increasing d. Empirically, we set $d = 60$ on Foursquare dataset and $d = 70$ on Twitter dataset in our experiments, which achieves a satisfying trade off between recommendation accuracy and efficiency.

Table 3. Impact of metric dimension d

(a) Impact of d on Foursquare

d	Accuracy@k				
	$k = 1$	$k = 5$	$k = 10$	$k = 15$	$k = 20$
30	0.160	0.221	0.249	0.267	0.281
40	0.263	0.317	0.340	0.354	0.366
50	0.363	0.417	0.428	0.450	0.461
60	**0.381**	**0.436**	**0.456**	**0.470**	**0.480**
70	0.382	0.436	0.456	0.471	0.480
80	0.383	0.436	0.456	0.471	0.481

(b) Impact of d on Twitter

d	Accuracy@k				
	$k = 1$	$k = 5$	$k = 10$	$k = 15$	$k = 20$
40	0.193	0.282	0.333	0.365	0.388
50	0.213	0.295	0.341	0.372	0.395
60	0.223	0.310	0.359	0.390	0.418
70	**0.233**	**0.315**	**0.363**	**0.396**	**0.420**
80	0.233	0.316	0.363	0.396	0.420
90	0.234	0.316	0.363	0.397	0.421

Impact of Time Period Threshold $\boldsymbol{\triangle T}$**.** Table 4 investigates the impact of time period threshold $\triangle T$ in GME-S. From the experimental results, we observe

Table 4. Impact of time period threshold $\triangle T$

(a) Impact of $\triangle T$ on Foursquare

$\triangle T$	Accuracy@k				
	$k = 1$	$k = 5$	$k = 10$	$k = 15$	$k = 20$
1	0.233	0.267	0.284	0.297	0.307
3	0.246	0.299	0.327	0.342	0.351
5	**0.381**	**0.436**	**0.456**	**0.470**	**0.480**
7	0.304	0.361	0.385	0.402	0.414
9	0.165	0.226	0.255	0.276	0.289
∞	0.146	0.176	0.194	0.209	0.222

(b) Impact of $\triangle T$ on Twitter

$\triangle T$	Accuracy@k				
	$k = 1$	$k = 5$	$k = 10$	$k = 15$	$k = 20$
10	0.216	0.278	0.315	0.339	0.359
15	0.226	0.302	0.345	0.373	0.395
20	**0.233**	**0.315**	**0.363**	**0.396**	**0.420**
25	0.230	0.311	0.356	0.386	0.408
30	0.212	0.295	0.322	0.361	0.388
∞	0.190	0.244	0.279	0.304	0.324

that the performance first improves quickly with the increase of $\triangle T$ and then drop down rapidly. Note that, when $\triangle T = \infty$, GME-S reduces to GME model. The reason of accuracy disparity is that, when $\triangle T$ is small, GME-S may prune too many POI co-occurrence edges which makes the train set too small to completely training, while $\triangle T$ is large, GME-S may incorporate too many noise edges, which may lead to lower accuracy in test set. Thus, we choose $\triangle T = 5$ days on Foursquare dataset and $\triangle T = 20$ days on Twitter dataset to get the best result. Moreover, due to the denser check-in data on Foursquare dataset compared to that on Twitter dataset, the $\triangle T$ is smaller on Foursquare dataset than that on Twitter dataset.

4 Related Work

In this section, we discuss existing research related to our work, including next POI recommendation and metric embedding.

Importance of POI recommendation has attracted a significant amount of research interest on developing recommendation techniques [12,18,20,23,24,26], while the next POI recommendation which requires providing satisfactory recommendations promptly based on users' latest preferences and their most recent checked-in POIs has received relatively little research attention. Most of the studies developed the Markov chain-based methods to capture the sequential patterns of POIs and predict the next check-ins. To reduce the size of the prediction space, Cheng et al. [2] exploited sequential influence using the first-order Markov chain which only considers the latest location in a user's visiting sequence to recommend a new location for the user. Zhang et al. [28] predicted the next location probability through an additive Markov chain, and assumed recent check-in locations usually have stronger influence than those locations checked-in long time ago. Although the parameter space in these approaches can be decreased to $|V| \times |V|$, it may still be huge considering that V is usually a large number in LBSNs. To reduce the prediction space, Wang et al. [16] modeled the sequential effect at the topic-region level. However, its accuracy fell behind us. In our proposed GME-S model, the parameter space is only $|V| \times d$, where d is the dimension of POI embedding that tends to be smaller than 100, thus the parameter space is much smaller than $|V| \times |V|$.

Embedding methods have been long studied and proved to be effective in capturing latent semantics of how items (e.g. words in sentences) interact with each other. For example, Tang et al. [14] learned words embedding to make document classification, and verified its effectiveness. There are also a line of music recommendation research using metric embedding based methods. Chen et al. [1] adopted metric embedding in the music playlist prediction and proposed a Logistic Markov embedding (LME) for generating the playlists. The research [17] proposed by Wu et al. embeds users and songs into a common latent space to represent the personalized Markov chain. The POI recommendation using metric embedding methods is relatively less. PRME proposed by Feng et al. [4] and BPR developed by Rendle et al. [12] are the typical ones which exploits pair-wise ranking scheme. However, our work is a graph-based method, which can embed large-scale information efficiently into a graph and represent POIs and users in a unified metric while PRME embeds user preference and sequential patterns in two different metric respectively. Moreover, we track and represent the dynamic user preferences in the form of time-decay which can make recommendation in a real-time manner based on users' latest preferences.

Our work in this paper distinguishes itself from previous researches in several aspects. Firstly, to the best of our knowledge, it is the first effort that uses the metric embedding method to unify dynamic user preferences and the sequential influence in a principled way. Secondly, although research [4] exploited the metric embedding for next POI recommendation, it embedded user preferences and sequential transition into two different spaces which may lose some potential relationship between users and POIs. In contrast, our proposed methods embed all the information into a unified space via graph based method and make next POI recommendation by tracking the change of user preferences in a real-time manner. Thirdly, we proposed a novel effective and efficient method to exploit and encode the knowledge of sequential patterns of users' check-in behaviors in the POI embedding and track the dynamics of user preferences in an efficient way.

5 Conclusions

In this paper, we proposed a novel graph-based metric embedding (GME) model for next POI recommendation which can learn the representation of POIs in a low-dimension latent space and track the dynamic user preferences in the form of time-decay based on the learnt POI embeddings. To model the sequential influence, we further extend our GME to GME-S model by exploiting and integrating the sequential patterns in the learning process of POI embeddings. To the best of our knowledge, this is the first work that uses the metric embedding method to unify dynamic user preferences and the sequential influence in a principled manner. Extensive experiments were conducted to evaluate the performance of GME and GME-S on two real datasets. The results showed superiority of our proposals over other competitor methods. Besides, we studied the impact of time interval of sequential patterns and verified the importance of sequential influence in next POI recommendation. According to the experimental results, our approach significantly outperforms existing recommendation methods in effectiveness and efficiency.

Acknowledgments. This work was supported by National Basic Research Program of China (2013CB329305), ARC Discovery Early Career Researcher Award (DE160100308) and ARC Discovery Project (DP140103171). It was also partially supported by National Natural Science Foundation of China (61572335, 61303164, 61402447 and 61502466), Development Plan of Outstanding Young Talent from Institute of Software, Chinese Academy of Sciences (ISCAS2014-JQ02) and Jiangsu Natural Science Foundation of China (BK20151223).

References

1. Chen, S., Moore, J.L., Turnbull, D., Joachims, T.: Playlist prediction via metric embedding. In: KDD, pp. 714–722 (2012)
2. Cheng, C., Yang, H., Lyu, M.R., King, I.: Where you like to go next: successive point-of-interest recommendation. In: IJCAI, pp. 2605–2611 (2013)
3. Cheng, Z., Caverlee, J., Lee, K., Sui, D.Z.: Exploring millions of footprints in location sharing services. In: ICWSM (2011)
4. Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y.M., Yuan, Q.: Personalized ranking metric embedding for next new poi recommendation. In: AAAI, pp. 2069–2075 (2015)
5. Ference, G., Ye, M., Lee, W.-C.: Location recommendation for out-of-town users in location-based social networks. In: CIKM, pp. 721–726 (2013)
6. Gao, H., Tang, J., Hu, X., Liu, H.: Exploring temporal effects for location recommendation on location-based social networks. In: RecSys, pp. 93–100 (2013)
7. Li, A.Q., Ahmed, A., Ravi, S., Smola, A.J.: Reducing the sampling complexity of topic models. In: KDD, pp. 891–900 (2014)
8. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2003)
9. Liu, B., Fu, Y., Yao, Z., Xiong, H.: Learning geographical preferences for point-of-interest recommendation. In: KDD, pp. 1043–1051 (2013)
10. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in NeuralInformation Processing Systems, pp. 3111–3119 (2013)
11. Recht, B., Re, C., Wright, S., Niu, F.: HOGWILD: a lock-free approach to parallelizing stochastic gradient descent. In: Advances in Neural Information Processing Systems, pp. 693–701 (2011)

12. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: UAI, pp. 452–461 (2012)
13. Tang, J., Meng, Q., Mei, Q.: PTE: predictive text embedding through large-scale heterogeneous text networks. In: KDD, pp. 1165–1174 (2015)
14. Tang, J., Meng, Q., Wang, M., Zhang, M., Yan, J., Mei, Q.: LINE: large-scale information network embedding. In: WWW, pp. 1067–1077 (2015)
15. Wang, W., Yin, H., Chen, L., Sun, Y., Sadiq, S., Zhou, X.: Geo-SAGE: a geographical sparse additive generative model for spatial item recommendation. In: KDD, pp. 1255–1264 (2015)
16. Wang, W., Yin, H., Sadiq, S., Chen, L., Xie, M., Zhou, X.: SPORE: a sequential personalized spatial item recommender system. In: ICDE (2016)
17. Wu, X., Liu, Q., Chen, E., He, L., Lv, J., Cao, C., Hu, G.: Personalized next-song recommendation in online karaokes. In: RecSys, pp. 137–140 (2013)
18. Ye, M., Yin, P., Lee, W.-C., Lee, D.-L.: Exploiting geographical influence for collaborative point-of-interest recommendation. In: SIGIR, pp. 325–334 (2011)
19. Yin, H., Zhou, X., Cui, B., Wang, H., Zheng, K., Nguyen, Q.V.H.: Adapting to user interest drift for poi recommendation. IEEE Trans. Knowl. Data Eng. **PP**(99), 1–14 (2016)
20. Yin, H., Cui, B.: Spatio-Temporal Recommendation in Social Media, 1st edn. Springer Publishing Company, Heidelberg (2016)
21. Yin, H., Cui, B., Chen, L., Zhiting, H., Zhang, C.: Modeling location-based user rating profiles for personalized recommendation. TKDD **9**(3), 19:1–19:41 (2015)
22. Yin, H., Cui, B., Huang, Z., Wang, W., Wu, X., Zhou, X.: Joint modeling of users' interests and mobility patterns for point-of-interest recommendation. In: ACM Multimedia, pp. 819–822 (2015)
23. Yin, H., Cui, B., Sun, Y., Zhiting, H., Chen, L.: LCARS: a spatial item recommender system. ACM Trans. Inf. Syst. **32**(3), 11:1–11:37 (2014)
24. Yin, H., Sun, Y., Cui, B., Zhiting, H., Chen, L.: LCARS: a location-content-aware recommender system. In: KDD, pp. 221–229 (2013)
25. Yin, H., Zhou, X., Shao, Y., Wang, H., Sadiq, S.: Joint modeling of user check-in behaviors for point-of-interest recommendation. In: CIKM, pp. 1631–1640 (2015)
26. Yuan, Q., Cong, G., Ma, Z., Sun, A., Thalmann, N.M.: Time-aware point-of-interest recommendation. In: SIGIR, pp. 363–372 (2013)
27. Zhang, J.-D., Chow, C.-Y.: Spatiotemporal sequential influence modeling for location recommendations: a gravity-based approach. TIST **7**(1), 11:1–11:25 (2015)
28. Zhang, J.-D., Chow, C.-Y., Li, Y.: LORE: exploiting sequential influence for location recommendations. In: SIGSPATIAL, pp. 103–112 (2014)
29. Zheng, Y.-T., Zha, Z.-J., Chua, T.-S.: Mining travel patterns from geotagged photos. TIST **3**(3), 56:1–56:18 (2012)

Temporal Pattern Based QoS Prediction

Liang Chen[1](✉), Haochao Ying[2], Qibo Qiu[2], Jian Wu[2], Hai Dong[1],
and Athman Bouguettaya[1]

[1] School of Computer Science and Information Technology,
RMIT, Melbourne, Australia
jasonclx@gmail.com, {hai.dong,athman.bouguettaya}@rmit.edu.au

[2] College of Computer Science and Technology, Zhejiang University,
Hangzhou, China
{haochaoying,vincent2014,wujian2000}@zju.edu.cn

Abstract. Quality-of-Service (QoS) is critical for selecting the optimal Web service from a set of functionally equivalent service candidates. Since QoS performance of Web services are unfixed and highly related to the service status and network environments which are variable against time, it is critical to obtain the missing QoS values of candidate services at given time intervals. In this paper, we propose a temporal pattern based QoS prediction approach to address this challenge. Clustering approach is utilized to find the temporal patterns based on services QoS curves over time series, and polynomial fitting function is employed to predict the missing QoS values at given time intervals. Furthermore, a data smoothing process is employed to improve prediction accuracy. Comprehensive experiments based on a real world QoS dataset demonstrate the effectiveness of the proposed prediction approach.

Keywords: Service Computing · QoS prediction · Temporal pattern

1 Introduction

A Service-Oriented Computing (SOC) paradigm and its realization through standardized Web service technologies provide a promising solution to the seamless integration of single-function applications to create new large-grained and value-added services. Web services are software systems designed to support interoperable machine-to-machine interaction over a network. Typically, a service-oriented application consists of multiple Web services interacting with each other in several tiers.

Quality of Service (QoS) has been widely employed for evaluating the non-functional characteristics of Web services [16]. With the explosive growth of functionality-equal services, non-functional characteristic of Web service is becoming a popular research concern and kinds of QoS-based approaches were proposed in various of Service Computing areas, such as service composition [1,2], fault-tolerant web services [5], and service selection [4,18].

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 223–237, 2016.
DOI: 10.1007/978-3-319-48743-4_18

A common premise of previous research is that the values of QoS properties are already known and fixed. However, user-dependent QoS values always vary over time in the real-world scenario. Figure 1(a)[1] shows the variation curve of one service's response time (response time is one important QoS property) when continually invoked by the same user along 64 time intervals. It could be found that the response time varies largely from 1 s to 20 s. Actually, the QoS performance of Web services observed from the users perspective is usually quite different from that declared by the service providers in Service Level Agreement (SLA), due to the following reasons [17]:

- QoS performance of Web services is highly related to invocation time, since the service status (e.g., workload, number of clients, etc.) and the network environment (e.g., congestion, etc.) change over time.
- Service users are typically distributed in different geographical locations. The user-observed QoS performance of Web services is greatly influenced by the Internet connections between users and Web services. Different users may observe quite different QoS performance when invoking the same Web service.

Based on above reasons, it is becoming essential to collect time-aware QoS information of Web services for QoS-based Service Computing research issues. However, in reality, a service user usually only invokes a limited number of Web services, thus the QoS values of the other Web services are missing (unknown) for the target user. Without sufficient time-aware QoS information, the accuracy of QoS-based research work, i.e., QoS-based service selection, QoS-base service composition, could not be guaranteed. Therefore, it is becoming urgent to build a time-aware QoS prediction approach for efficiently estimating missing QoS values of Web services for target users.

In this paper, we propose to address the problem of time-aware QoS prediction by exploring the advantages of temporal patterns. Temporal patterns and related techniques have been used and demonstrated in social media area to solve the problems such as video popularity prediction in Youtube [12], retweet number prediction in Twitter [15], etc. An intuitive idea is that the influences of factors (i.e., network environment, location, etc.) behind the QoS temporal variation could be reflected in the uncovered patterns, and the missing values in each QoS carve could be predicted by using the most similar temporal patter to fit for. Particularly, a curve clustering approach is proposed to uncover QoS temporal patterns, and polynomial fitting function is employed to predict the missing QoS values. Moreover, A curve smoothing approach is employed to improve prediction accuracy, due to the noises in QoS curves. Experiments based on 20+ million service invocation records demonstrate the effectiveness of the proposed prediction approach.

In summary, this paper makes the following contributions:

1. We formally identify the critical problem of time-aware Web service QoS prediction and propose the concept of temporal pattern in this research area. Particularly, temporal patterns are extracted from QoS curves over time series.

[1] Due to the space limitation, Fig. 1 is placed in Page 5.

2. We propose a novel Temporal Pattern based QoS Prediction approach TPP, which utilizes temporal patterns to predict the missing QoS values via polynomial fitting. Moreover, a data smoothing process is employed to improve the prediction accuracy. We consider TPP as the first temporal pattern based QoS prediction approach.
3. Comprehensive experiments based on a real world Web service QoS dataset are implemented to evaluate the performances of TPP and other state-of-the-art approaches. Compared with other approaches, TPP achieves 35.8 %∼52.0 % improvement in terms of MRE metric.

The rest of this paper is organized as follows. Section 2 highlights the related work of QoS prediction. Section 3 formally define the problem and introduces the details of data smoothing, pattern clustering, and the prediction algorithm. Experimental results and analysis are presented in Sect. 4, whereas Sect. 5 concludes this paper.

2 Related Work

Quality of Service (QoS) has been widely employed for evaluating the non-functional characteristics of Web services [16]. Among QoS properties, values of server-side QoS (e.g., price, popularity) are identical for different users while others (e.g., response time, throughput) observed from the user-side may change over time due to the unpredictable network conditions and heterogeneous user environments [8]. With the explosive growth of functionality-equal services, non-functional characteristic of Web service is becoming a popular research topic and kinds of QoS-based approaches are proposed in various of Service Computing areas, such as service composition service composition [1] fault-tolerant web services [5] and service selection [4].

A common premise of previous research is that the values of user-dependent QoS properties are already known. However, in reality a user typically has engaged a limited number of Web services in the past and cannot exhaustively invoke all the available candidate services. Thus, it is fundamental to predict the missing QoS values for any QoS-based Service Computing research.

In web service QoS prediction, Collaborative filtering approaches have been widely adopted. Generally, traditional recommendation approaches could be classified into two categories: memory-based [13,19] and model-based [3]. Memory-based approaches, also known as neighborhood-based approaches, are one of the most popular prediction methods in collaborative filtering systems. Shao et al. [11] first use collaborative filtering approach to predict QoS values from similar users. Zheng et al. [20] propose a hybrid user-based and item-based approach to predict QoS values for the current user by employing historical web service QoS data from other similar users and similar web services. Although memory-based algorithms implement easily, high computation complexity makes it difficult to deal with a large and sparse time-aware dataset. Model-based algorithms employ statistical and machine learning techniques to learn a sophisticated model based on history QoS invocation records, including clustering models [14], latent semantic models [6], latent factor models [9], etc. Zheng et al. use PMF algorithm to predict missing failure probability values in user-service matrix [19], and propose NIMF to improve prediction accuracy by balancing the global information and local information [21]. Compared with memory-based approaches, model-based QoS prediction approaches usually have better performance but lack of interpretation.

Time is an important context factor which affects QoS prediction accuracy, since service status (e.g. number of clients and workload) and network environments (e.g. congestion) change over time. QoS values will fluctuate when the same user invoke the same service at different time interval. Limited QoS prediction works consider the influence of time to QoS values. Hu et al. propose a time-aware similarity model which considers two aspects: (1) More temporally close QoS experience from two users on a same service contributes more to the user similarity measurement; (2) More recent QoS experience from two users on a same service contributes more to the user similarity measurement [7]. Zhang et al. construct a three dimensional matrix by adding time factor, and then employ tensor factorization to extract user-specific, service-specific, and time-specific latent features from historical QoS values for prediction [17]. In this paper, we take advantage of model-based concept and propose a temporal pattern based approach with better interpretability. In this paper, we analyze a set of 430,000 response-time curves, each curve means one user invokes one service at 64 continuous time intervals. The surprising thing is that temporal patterns of QoS values could be accurately represented by using limited number of curves. Moreover, a data smoothing process is employed to improve the performance of QoS prediction.

3 QoS Prediction Based on Temporal Patterns

In this section, we first formally define the problem and analyze the research challenges in Sect. 3.1, and then introduce the details of corresponding solutions in Sects. 3.2 and 3.3, respectively. Finally, QoS prediction algorithm is presented in Sect. 3.4.

3.1 Problem Definition and Research Challenges

In previous works, most of QoS prediction approaches origin from recommender system. Concretely, they predict the missing QoS values in the user-service or user-service-time QoS matrix, which is generated from historical service invocation by users [11,17,22]. Unlike above works, we propose a novel method to predict QoS value based on temporal patterns in this paper.

Let U be the set of m users, S be the set of n Web services, and T be the set of c time intervals. From the collection of QoS attribute from user-side, the observed QoS value of user i invoking service j at time interval t_k can be formally represented by q_{ijk}, where $i \in 1, ..., m$, $j \in 1, ..., n$, $k \in 1, ..., c$ and q_{ijk} is one of QoS attributes (e.g., response time or throughput). For convenience, the length of time internals is fixed. For example, the real-world dataset employed in this paper is over 64 consecutive time slices at 15 min interval. Intuitively, the shape of q_{ij} measures how user i invokes service j changed over time. In practice, each user typically uses a few of services so that we can get the set H of complete curves which we know all QoS values of user invoked service at each time interval. However, component service can be replaced automatically in service-oriented architecture (SOA). Therefore, the records of user invoked service at some intervals may be missed, which formally represented by the set Δ. Our goal is to use the complete curves in H to predict the missing value in Δ.

In the scenario of QoS prediction, there are two challenges to efficiently predict the missing value. First, due to the influence of dynamic network conditions and varying server loads, the QoS value at each interval fluctuates quickly and may exist noise. If we directly use the original data, the performance of prediction may reduce. Secondly, To predict the missing value, the naive method is to compare the curve with missing value with each complete curve and then use the most similar curve to predict the missing value. However, although each user invokes a few services, millions of complete curves may be collected if we have large number of users and services. Therefore, this approach is time-consuming and not efficiently.

3.2 Data Smoothing

To deal with the first challenge, we design a data transformation method for QoS data to reduce noises. Figure 1 presents an example of a complete QoS (i.e., response time) curve of one user invoked a service. It is obvious that the curve is too diverse to directly compare with others by using distance measure. Fortunately, We can also observe that the QoS value at time t is close to the value at the previous $(t-1)$ and forward $(t+1)$ time slice. It is intuitive that

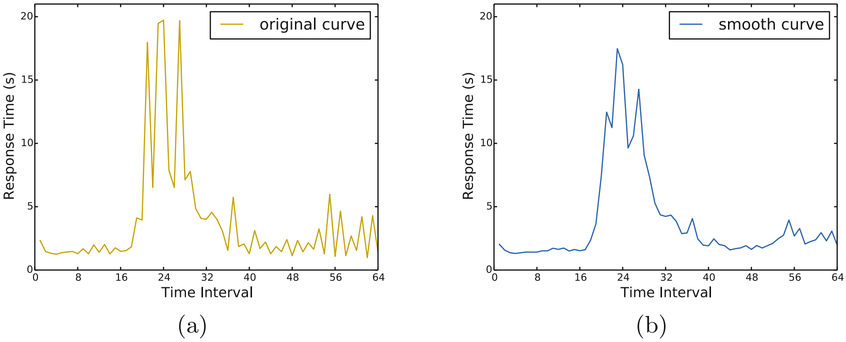


Fig. 1. An example of curves smooth in response time. (a) is the original curve and QoS values fluctuate sharply. (b) is the smoothed curve. To some extent this curve reduces noise and keep overall changing shape of QoS values.

the value in continue intervals should be similar. Based on this observation, the data transformation is defined as follows:

$$q_{ijk} = \begin{cases} \dfrac{2q_{ijk} + q_{ijk+1}}{3} & k = 1 \\ \dfrac{2q_{ijk} + q_{ijk-1}}{3} & k = c \\ \dfrac{q_{ijk-1} + 2q_{ijk} + q_{ijk+1}}{4} & \text{otherwise} \end{cases} \tag{1}$$

Our data smoothing method takes more weights to current observed QoS values and simultaneously consider QoS values in adjacent time. From Fig. 1(b), we can find that the smoothed curve retains the changing shape of QoS values and reduces noise to some extent.

3.3 Temporal Pattern Generation

To deal with the second challenge, we employ K-Means clustering algorithm to find the clusters of QoS curves that share distinct temporal pattern. The reason that we choose K-Means algorithm is its simpleness and efficiency.

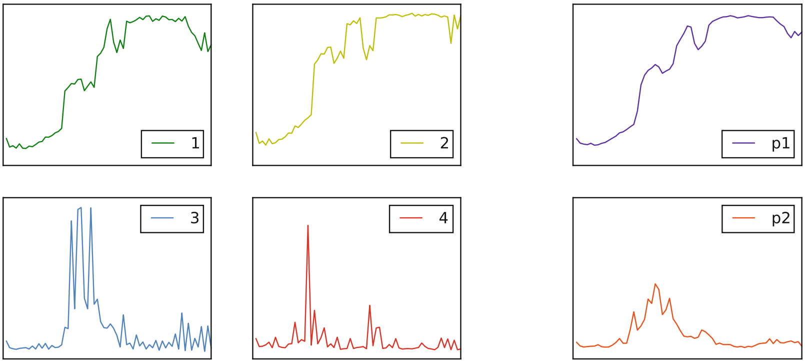
Given the set H of complete QoS curves and the number of clusters K, our goal is to find an assignment set C_k of curves for each cluster, and the centroid u^k of each cluster minimizes the following function:

$$F = \sum_{k=1}^{K} \sum_{q_{ij} \in C_k} d(q_{ij,} u^k) \tag{2}$$

where $d(q_{ij,} u^k) = \sum_{t=1}^{c} (q_{ijt}, u_t^k)^2$ is the square of Euclidean distance. We start the K-Means algorithm with random initial K centroids. As an iterative refinement algorithm, K-Means proceeds by alternating between two steps: assignment step and update step. In the assignment step, we assigns each curve to the cluster with the closest centroid based on $d(q_{ij,} u^k)$. After finding the new assignment set C_k for each curve, we calculate the new centroid for each C_k in the update step, according the average of all curves in C_k. Formally, the updated centroid should be as follows:

$$u^k = \frac{1}{|C_k|} \sum_{q_{ij} \in C_k} q_{ij} \tag{3}$$

After updating many times, the algorithm will converge when the assignment no longer changes. Finally, the centroid of each cluster represents the temporal pattern. Figure 2 presents an example for clustering four original curves. It is obvious that the two temporal patterns catch the most important characters in each cluster. In the next section, we will use these temporal patterns to predict the missing QoS values.



(a)

(b)

Fig. 2. An example of curves clustering. (a) is the four original curves of response time. After smoothing and clustering, (b) shows the centroids of two clusters.

3.4 QoS Prediction

After smoothing and clustering, we get K temporal patterns. Suppose that we have a curve q_{ij} with some QoS values at the corresponding time intervals missing. For simplify, q_{ij} misses the value in interval t. Now, the question is how to use these pattern to predict the missing QoS value q_{ijt}.

First, we compute the distance between the observed values of q_{ij} and each pattern in corresponding time interval under different metric. In the experiments, we compare three distance approaches (i.e., cosine, euclidean, and cityblock) and choose the best metric to measure the distance. After this step, the most similar pattern p can be obtained based on the distance. An intuitive way is to directly use the value of p in interval t to predict the q_{ijt}. However, it is unwise because the pattern p can not match q_{ij} completely and we can not eliminate the fixed distance in interval t. In this paper, we use a function to map the pattern p to the curve q_{ij}. In general, the map function is polynomial fitting function as follows:

$$\hat{q}_{ij} = map(p) = w_0 + w_1 p + w_2 p^2 + \cdots + w_d p^d \tag{4}$$

where w is the weights and d is the order.

After finding the order and weights of polynomial based on sum of least square between q_{ij} and $\hat{q}_{ij}$ in the observed values, the predicted value q_{ijt} could be obtained through $map(p_t)$. Note that we just compare the linear and square fitting in the experiments to avoid overfitting. The pseudo code of our algorithm for QoS prediction is provided in Algorithm 1.

4 Experiments

In this section, comprehensive experiments are implemented to evaluate the proposed approach based on a real-world dataset. Experimental evaluation will

Algorithm 1. Our QoS Prediction Algorithm.

```
Input : The set H of complete QoS curves; The number of clusters $K$; The set $\Delta$ of incomplete QoS curves
Output: The QoS prediction of unobserved value in $\Delta$
for $q \in \mathrm{H}$ do
    smooth q by Equation 1;
random initial $K$ centroids $u^1, u^2, ..., u^K$;
repeat
    set each cluster $C_1, .., C_K$ to null;
    for $j = 1$ to $|\mathrm{H}|$ do
        $k \longleftarrow argmin_{k=1,..,K} d(p_j, u^k)$;
        $C_k \longleftarrow C_k \cup j$;
    for $i = 1$ to $K$ do
        $u_i \longleftarrow \frac{1}{|C_k|} \sum_{q_{ij} \in C_k} q_{ij}$;
until centroids converge;
for $q \in \Delta$ do
    find the most similar pattern $p$ based on observed value in $q$;
    polynomial fit $q \longleftarrow map(p)$;
    predict unobserved value of $q$ in each interval $t$ by $map(p_t)$;
```

answer the following questions: (1) What are the evaluation metrics? (2) How does our approach compare with other state-of-the-art ones? (3) What is the impact of data smoothing, similarity approach, and the order of polynomial fit?

4.1 Data Preprocessing

In the experiments, we mainly focus on Response Time (RT), one of the most important QoS properties, to evaluate QoS prediction methods. Response time (RT) is the length of time between the end of an inquiry on a computer system and the beginning of a response. All experiments are implemented in a machine with a 2.2 GHz Intel CPU and 16 GB RAM, running OS X Yosemite.

For the sake of application in practice, all experiments are implemented based on a public real-world Web service QoS dataset which is collected by 142 users invoking 4532 web services in 16 hours with a time interval of 15 min [17]. In particular, the users are 142 computers of PlanetLab[2] located in 22 countries, and the services are 4532 public available real world web services distributed in 57 countries. Through the observation, we find quite a lot of noises exist in the dataset. For example, the response time value will be set to -1, if the response time is over 20 s in this invocation. Furthermore, some Web services have not been invoked by any user. Thus, we do some data cleaning work on this dataset, and macroscopic statistics & data distribution of the generated dataset

[2] PlanetLab is a global research network that supports the development of new network services. Details could be found in https://www.planet-lab.org/.

are presented in Figs. 3 and 4, respectively. It could be found the experimental evaluation utilizes more than 20 million records, which partly demonstrate reliability and scalability of the experiments. It should be noted that the proposed approach could be utilized for the prediction of any other QoS property (e.g., throughput), even though only response time is studied in this paper.

Statistics	Values
#Users	135
#Services	3952
#Time slices	64
#Time interval	15min
#Records	20,138,880
RT scale	$(0, 20)$
RT mean	0.8442

Fig. 3. Statistics of QoS dataset

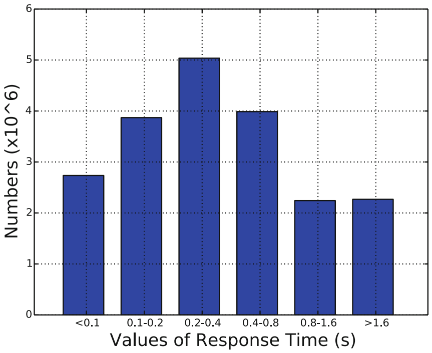


Fig. 4. RT value distribution

4.2 Evaluation Metric

We evaluate the prediction accuracy of our proposed approach in comparison with other existing methods by using the following metrics.

- **MAE** (Mean Absolute Error). MAE is average prediction accuracy between prediction results and corresponding observations, which is defined as follows:

$$MAE = \frac{\sum_{i,j} \left| \hat{R}_{ij} - R_{ij} \right|}{N} \tag{5}$$

  where R_{ij} denotes the real QoS value of service j observed by user i, $\hat{R}_{ij}$ is the predicted QoS value by a method, and N is the total number of predicted values.
- **NMAE** (Normalized Mean Absolute Error). NMAE normalizes the differences range of MAE by computing:

$$NMAE = \frac{MAE}{\sum_{ij} R_{ij}/N} \tag{6}$$

- **MRE** (Median Relative Error). MRE measures the median value of relative errors between observed value and predicted value:

$$MRE = median \left| \hat{R}_{ij} - R_{ij} \right| / R_{ij} \tag{7}$$

Due to the large variance of QoS values, we focus more on relative error metric, i.e., MRE, which is more appropriate for QoS prediction evaluation. Since many papers use MAE and NMAE, they are also included for comparison purpose.

4.3 Performance Comparisons

In order to show the effectiveness of our proposed QoS prediction approach, we compare the prediction accuracy of the following methods:

- **UPCC**: This method employs the information of similar users (measured by Pearson Correlation Coefficient) to predict the QoS values [3].
- **IPCC**: This method is widely-used in recommendation system, which employs the similarity between services for QoS prediction [10].
- **UIPCC**: This method combines UPCC and IPCC model, which fully uses the similarity of users and services [20].
- **PMF**: This is a classic matrix factorization method, which has been employed in [19]. User-service matrix is factorized into two matrices under low-rank assumption and then using the matrices predict QoS values.
- **WSPred**: This is a tensor factorization-based prediction method with average QoS value constraint [17].

In the experiments, user-service records are randomly divided into two parts: 80 % records as the training data and the rest 20 % as the testing data. In order to evaluate the performance of different approaches in reality, we randomly choose $\frac{m}{16}$ ($m = 1, 2, 3, 4, 5, 6, 7, 8$) of the training data for pattern clustering, and the others (i.e., $\frac{16-m}{16}$ of the training data) for cross validation. Equation (1) is employed for data smoothing, and Eqs. (2) and (3) are employed for the pattern clustering. Through the observation of experimental results, we find that the proposed approach could get similar temporal patterns in any density setup. That is, the proposed pattern clustering approach is quite stable and even $\frac{1}{16}$ of training data is enough to get appropriate patterns. Polynomial fit is employed for QoS prediction, once temporal patters are generated. Since a user usually only invokes a small number of services, the testing matrix density is randomly thinned to the same $\frac{m}{16}$. The prediction accuracy is evaluated by comparing the original value and the predicted value of each removed entry in testing matrix. Without lost of generality, the number of patterns is set as 4 in this paper. Detailed impact of data smoothing, similarity approach, and polynomial order is studied in Sects. 4.4, 4.5, and 4.6, respectively.

The QoS value prediction accuracies evaluated by MAE, NAME, and MRE are shows in Table 1. For each row in the table, we highlight the best performer among all methods. As we can observe, our approach significantly outperforms the other ones over MRE, while still achieving best results on MAE and NMAE. Concretely, our approach achieves 35.8 %∼52.0 % improvement on MRE, 1.8 %∼2.7 % improvement on MAE, and 2.0 %∼3.0 % improvement on NMAE at different matrix densities. Note that all improvements are computed as the percentage of how much our approach outperforms the other most competitive approach.

Table 1. Comparison of performance (a smaller value means a better performance)

Method	Density = 2/16			Density = 3/16			Density = 4/16		
	MAE	NMAE	MRE	MAE	NMAE	MRE	MAE	NMAE	MRE
UPCC	0.5226	0.6211	0.5334	0.492	0.5845	0.477	0.4745	0.5637	0.4497
IPCC	0.5946	0.7066	0.6671	0.5675	0.6741	0.6395	0.5376	0.6386	0.5992
UIPCC	0.5215	0.6197	0.5225	0.4912	0.5835	0.473	0.4719	0.5606	0.4467
PMF	0.5219	0.6208	0.4764	0.4925	0.5855	0.4496	0.4765	0.5659	0.4327
WSPred	0.4583	0.5445	0.4519	0.4358	0.5168	0.4293	0.4253	0.504	0.4112
TPP	**0.4501**	**0.532**	**0.2167**	**0.4253**	**0.5025**	**0.2249**	**0.4138**	**0.4888**	**0.2308**
Improve. (%)	1.8 %	2.3 %	52.0 %	2.4 %	2.8 %	47.6 %	2.7 %	3.0 %	43.9 %
Method	Density = 5/16			Density = 6/16			Density = 7/16		
	MAE	NMAE	MRE	MAE	NMAE	MRE	MAE	NMAE	MRE
UPCC	0.462	0.549	0.4323	0.4517	0.5368	0.4185	0.4435	0.5272	0.4069
IPCC	0.5204	0.6184	0.5776	0.5071	0.6029	0.5606	0.4954	0.5891	0.5453
UIPCC	0.4588	0.5452	0.4307	0.4482	0.5327	0.4182	0.4394	0.5223	0.4072
PMF	0.4633	0.55	0.4262	0.4536	0.5386	0.4231	0.4444	0.5277	0.408
WSPred	0.4148	0.4913	0.3895	0.4125	0.4884	0.3894	0.4084	0.4834	0.3814
TPP	**0.4075**	**0.4817**	**0.2375**	**0.4026**	**0.4756**	**0.2419**	**0.3985**	**0.4709**	**0.2448**
Improve. (%)	1.8 %	2.0 %	39.0 %	2.4 %	2.6 %	37.9 %	2.4 %	2.6 %	35.8 %

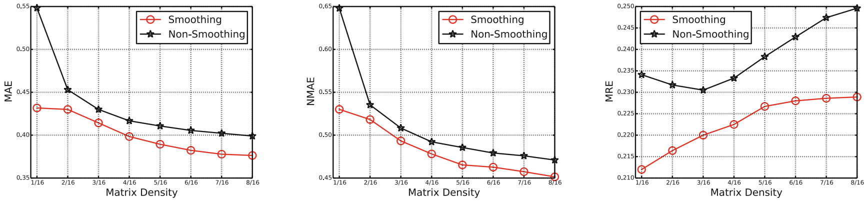
We also find that although UIPCC achieves higher accuracy than UPCC and IPCC over MAE and NMAE, and WSPred achieves better performance compared with the first three Collaborative Filtering based approaches (i.e., UPCC, IPCC, and UIPCC) and PMF, all these approaches have large errors over MRE. Thus, only focusing on minimizing the absolute error may lead to large relative error, which is not suitable for QoS prediction problem.

4.4 Impact of Data Smoothing

Data smoothing process is employed to reduce noises in QoS curves for the purpose of improving prediction accuracy, and is one of main contributions in this paper. In order to study its impact, we implement two versions of our proposed approach: one with the proposed data smooth process, i.e., Eq. (1), and the other without it. Figure 5 shows the prediction accuracy comparison between the above two versions. From Fig. 5, We can observe that the version with data smoothing largely outperforms the other version in terms of MAE, NMAE, and MRE. This is because the remove of noise points in QoS curves facilitates the generation of temporal patterns. In short, The process smooths out data fluctuations and improves QoS prediction accuracy.

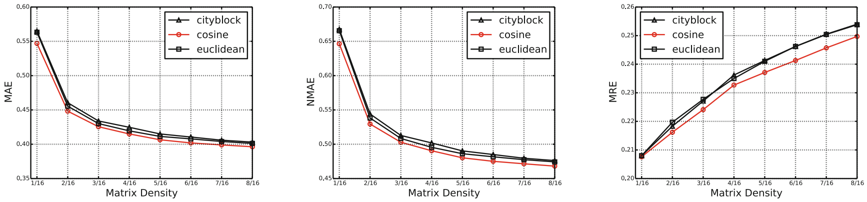
4.5 Impact of Similarity Approach

In the process of the proposed TPP approach, we have to choose the most similar pattern for the target QoS curve for predicting the missing values in



(a) (b) (c)

Fig. 5. Impact of data smoothing



(a) (b) (c)

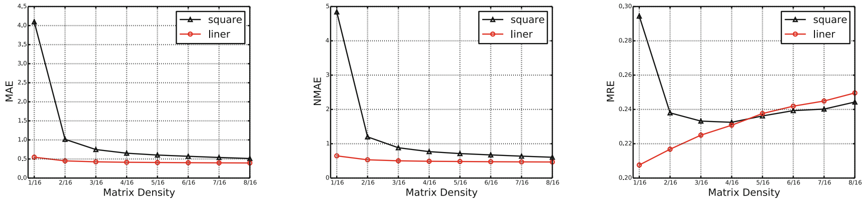
Fig. 6. Impact of similariy approach

the curve. Thus, the choice of similarity measure approach is very important for the final prediction accuracy. In the experiments, we employ three widely accepted approaches to compute the similarity between pattern and the choose data points in testing data. Specifically, the three similarity measure approaches are cosine, euclidean, and cityblock.

To present a comprehensive evaluation of these approaches, we vary the matrix density from 1/16 to 8/16. Other parameter settings are #pattern = 4, order of polynomial = 1. Figure 6 shows the performance comparison of different similarity approaches in terms of MAE, NAME, and MRE. From Fig. 6, we can find cosine similarity method always outperforms the other methods over three metrics when the data density varies from 1/16 to 8/16. This observation demonstrates that cosin similarity measurement is more suitable for computing similarity between curves. Furthermore, we can also observe that as the density increases, every similarity approach can achieve better prediction results in terms of absolute error metrics, i.e., MAE and NMAE. This is because more data points provided in testing data, more information could be gained for prediction. However, it is not suitable for the trend of relative error, i.e., MRE.

4.6 Impact of Order of Polynomial Fit

Once the optimal pattern is selected, polynomial fitting function is employed to predict the missing QoS values in testing data. In this section, we evaluate the impact of different polynomial fitting functions, that is, order of polynomial fit. For simplicity, we only compare the performance of QoS prediction when order



(a) (b) (c)

Fig. 7. Impact of order of polynomial fit

is 1 (liner) and 2 (square), since the trend could be easily illustrated by this comparison. Other parameter settings are #pattern = 4, similarity = cosine.

Figure 7 shows the prediction accuracy comparison of linear and square polynomial fit with the increase of density. For Fig. 7, we observe that the linear polynomial fit outperforms the square in most cases. As the increase of density, the prediction accuracy of square polynomial fit improves (the MAE and NMAE decreases) due to more information provided. However, compared with square one, it could be observed that linear polynomial fit is quite stable with the increase of density. That means, linear polynomial fit approach is very suitable for the case of cold-start and data sparsity, that is, online QoS prediction.

Further, we can find the prediction accuracy of square polynomial fit is quite bad when the density is 1/16, which means this sparsity condition causes an overfitting problem. From another perspective, the performance gap of linear and square polynomial fit decreases with the increase of matrix density. That means the overfitting phenomenon alleviates with more provided information. In all, linear polynomial fit is quite suitable for our problem.

5 Conclusion

With the explosive growth of functionality-equal services, non-functional characteristic of Web service is becoming a popular research concern and kinds of QoS-based approaches were proposed in various of Service Computing research areas. Since QoS performance of Web services are unfixed and highly related to the service status and network environments which are variable against time, it is critical to obtain the missing QoS values of candidate services at given time intervals. In this paper, we propose a temporal pattern based QoS prediction approach to address this challenge. Clustering approach is utilized to find the temporal patterns based on services QoS curves over time series, and polynomial fitting function is employed to predict the missing QoS values at given time intervals. Furthermore, a data smoothing process is employed to improve prediction accuracy. Comprehensive experiments based on a real world QoS dataset demonstrate the effectiveness of the proposed prediction approach.

For future work, we will investigate more techniques to improve the performance of temporal pattern generation and QoS prediction. Particularly, QoS curve shifting and scaling techniques will be introduced for better pattern generation, and machine learning techniques will be utilized to predict the missing QoS values based on the generated temporal patterns. Further, the datasets of other QoS properties (e.g., throughput) will also be employed to evaluate the performance of the proposed approach.

Acknowledgement. This research was made possible by NPRP 9-224-1-049 grant from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the authors. This research was partially supported by the Natural Science Foundation of China under grant of No. 61379119, Science and Technology Program of Zhejiang Province under grant of No. 2013C01073, the Open Project of Qihoo360 under grant of No. 15-124002-002.

References

1. Alrifai, M., Risse, T.: Combining global optimization with local selection for efficient QoS-aware service composition. In: Proceedings of the 18th International Conference on World Wide Web, pp. 881–890. ACM (2009)
2. Alrifai, M., Risse, T., Nejdl, W.: A hybrid approach for efficient web service composition with end-to-end QoS constraints. ACM Trans. Web (TWEB) **6**(2), 7 (2012)
3. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, pp. 43–52. Morgan Kaufmann Publishers Inc. (1998)
4. Chen, L., Kuang, L., Wu, J.: Mapreduce based skyline services selection for QoS-aware composition. In: 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), pp. 2035–2042. IEEE (2012)
5. Fang, C.L., Liang, D., Lin, F., Lin, C.C.: Fault tolerant web services. J. Syst. Archit. **53**(1), 21–38 (2007)
6. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Trans. Inf. Syst. (TOIS) **22**(1), 89–115 (2004)
7. Hu, Y., Peng, Q., Hu, X.: A time-aware and data sparsity tolerant approach for web service recommendation. In: 2014 IEEE 21th International Conference on Web Services (ICWS), pp. 33–40. IEEE (2014)
8. Menasce, D.: QoS issues in web services. IEEE Internet Comput. **6**(6), 72–75 (2002)
9. Mnih, A., Salakhutdinov, R.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems, pp. 1257–1264 (2007)
10. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, pp. 175–186. ACM (1994)
11. Shao, L., Zhang, J., Wei, Y., Zhao, J., Xie, B., Mei, H.: Personalized QoS prediction for web services via collaborative filtering. In: IEEE International Conference on Web Services, ICWS 2007, pp. 439–446. IEEE (2007)
12. Szabo, G., Huberman, B.A.: Predicting the popularity of online content. Commun. ACM **53**(8), 80–88 (2010)
13. Wu, J., Chen, L., Feng, Y., Zheng, Z., Zhou, M.C., Wu, Z.: Predicting quality of service for selection by neighborhood-based collaborative filtering. IEEE Trans. Syst. Man Cybern.: Syst. **43**(2), 428–439 (2013)

14. Xue, G.R., Lin, C., Yang, Q., Xi, W., Zeng, H.J., Yu, Y., Chen, Z.: Scalable collaborative filtering using cluster-based smoothing. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 114–121. ACM (2005)
15. Yang, J., Leskovec, J.: Patterns of temporal variation in online media. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 177–186. ACM (2011)
16. Zeng, L., Benatallah, B., Ngu, A.H., Dumas, M., Kalagnanam, J., Chang, H.: QoS-aware middleware for web services composition. IEEE Trans. Softw. Eng. **30**(5), 311–327 (2004)
17. Zhang, Y., Zheng, Z., Lyu, M.R.: WSPred: a time-aware personalized QoS prediction framework for web services. In: 2011 IEEE 22nd International Symposium on Software Reliability Engineering (ISSRE), pp. 210–219. IEEE (2011)
18. Zhao, L., Ren, Y., Li, M., Sakurai, K.: Flexible service selection with user-specific QoS support in service-oriented architecture. J. Netw. Comput. Appl. **35**(3), 962–973 (2012)
19. Zheng, Z., Lyu, M.R.: Personalized reliability prediction of web services. ACM Trans. Softw. Eng. Methodol. (TOSEM) **22**(2), 12 (2013)
20. Zheng, Z., Ma, H., Lyu, M.R., King, I.: QoS-aware web service recommendation by collaborative filtering. IEEE Trans. Serv. Comput. **4**(2), 140–152 (2011)
21. Zheng, Z., Ma, H., Lyu, M.R., King, I.: Collaborative web service QoS prediction via neighborhood integrated matrix factorization. IEEE Trans. Serv. Comput. **6**(3), 289–299 (2013)
22. Zhu, J., He, P., Zheng, Z., Lyu, M.R.: Towards online, accurate, and scalable QoS prediction for runtime service adaptation. In: 2014 IEEE 34th International Conference on Distributed Computing Systems (ICDCS), pp. 318–327. IEEE (2014)

Searching for Data Sources for the Semantic Enrichment of Trajectories

Luiz André P. Paes Leme[1](✉), Chiara Renso[2], Bernardo P. Nunes[3,4], Giseli Rabello Lopes[5], Marco A. Casanova[3], and Vânia P. Vidal[6]

[1] Fluminense University, Niterói, RJ, Brazil
lapaesleme@ic.uff.br
[2] ISTI-CNR, Pisa, PI, Italy
chiara.renso@isti.cnr.it
[3] PUC-Rio, Rio de Janeiro, RJ, Brazil
{bnunes,casanova}@inf.puc-rio.br
[4] Federal University of the State of Rio de Janeiro, Rio de Janeiro, RJ, Brazil
bernardo.nunes@uniriotec.br
[5] Federal University of Rio de Janeiro, Rio de Janeiro, RJ, Brazil
giseli@dcc.ufrj.br
[6] Federal University of Ceará, Fortaleza, CE, Brazil
vvidal@lia.ufc.br

Abstract. The fast growing number of datasets available on the Web inspired researchers to propose innovative techniques to combine spatio-temporal data with contextual data. However, as the number of datasets has increased relatively fast, finding the most appropriate datasets for enrichment also became extremely difficult. This paper proposes an innovative approach to rank a set of datasets according to the likelihood that they contain relevant enrichments. The approach is based on the intuition that the sequence of places visited during a trajectory can induce the best datasets to enrich the trajectory. It relies on a supervised approach to learn rules of association between visited places and meaningful datasets.

Keywords: Trajectories · Semantic enrichment · Movement data

1 Introduction

The personal position-enabled mobile devices are becoming our companions in everyday life, leaving tracks of our movements during our daily routine. The tracks collected by mobile devices describe the so-called *raw trajectories* that represent the geometric facets of movement data. Social media have also been proposed as complementary sources of mobility data. Georeferenced social media can be used as sparse and freely annotated movement traces [2,12] or, possibly, can be used to enrich raw GPS data thus getting semantically richer data with high positional accuracy [5].

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 238–246, 2016.
DOI: 10.1007/978-3-319-48743-4_19

The approach presented in this paper tackles the problem of searching the most appropriate datasets to enrich mobility data. It is based on the intuition that the sequence of places visited during a movement, i.e., the sequence of stops, can induce the purpose of the movement and hence suggest the set of datasets for enrichment. For example, assume that a traveler visits the sequence of places `[hotel, stadium, restaurant, hotel]` in Rio de Janeiro. Also assume that the dataset of tourist attractions available in the Open Data Portal of the government of the city of Rio de Janeiro contains data about the Maracanã stadium. The sequence of places suggests that the person can be a tourist because tourists frequently stay in hotels and visit the Maracanã Stadium in Rio de Janeiro. Therefore, one could attempt to match the place labeled as `stadium` with the entry *Maracanã stadium* in the dataset. It is important to notice that this is not a deterministic problem that could be solved with an a priori rule such as `if a person visited a stadium then search for enrichments in the dataset of attractions` since there is no obvious evidence, for someone who doesn't know the content of the dataset, that the dataset of attractions would contain an entry that could be matched with the place `stadium`. However, this can be learned from previous trajectory enrichments: if most trajectories similar to this one, in terms of the places visited, are enriched with the dataset of attractions then one can select that dataset as a potential source of enrichment for the new trajectory.

In this paper we take advantage of social media traces of movement and their user annotations to propose a technique for searching potentially useful datasets for the enrichment of trajectories. As for related work, the process of semantic enrichment of spatial and spatiotemporal data can be automatic [2,5,10] or semi-automatic [8]. Automatic approaches can use machine learning techniques such as Hidden Markov Models [10,12], probabilistic models [7], similarity measures [2] or simple proximity heuristics [2] to attach annotations. Recent techniques have also stressed the relevant role of the emerging and fast growing Web of Data [3] in the enrichment process. All existing works have used predefined sets of sources. Developers have favored popular sources such as DBpedia, Open Street Map, Open Weather Map, etc. and neglected less popular ones such as government open data and domain specific datasets. The fundamental reason for that is the lack of techniques to crawl and search for potentially useful datasets for enrichment.

The rest of this paper is organized as follows. Section 2 introduces the basic concepts used throughout the paper and describes the proposed ranking technique. Section 3 addresses the preparation of the test dataset. Section 4 presents the experiments for assessing the technique and Sect. 5 contains the conclusions.

2 The Problem of Searching for Sources of Enrichments

A *raw trajectory* of a moving object o is a sequence $\rho_o = (p_1, p_2, ..., p_n)$ of spatio-temporal points such that the timestamp of p_i is earlier than the timestamp of p_{i+1}. A *segment* g of a raw trajectory ρ_o is a continuous subsequence of ρ_o.

A *segmented trajectory* of a raw trajectory ρ_o is a sequence $\sigma_o = (g_1, g_2, ..., g_n)$ of segments of ρ_o such that $s = g_1 \| ... \| g_n$, that is, s is the concatenation of $g_1, ..., g_n$. A segment of a raw trajectory is a fragment of the whole raw trajectory where a given property holds.

The notion of semantic trajectory goes further and enriches a segmented trajectory with contextual information retrieved from external datasets. A *contextual resource* r of a dataset d is a pair (r, d) with $r \in d$. We use the notion r^d rather than (r, d). A *contextual information* of a segment g of a segmented trajectory σ_o, denoted by c, is a set, of contextual resources $c = \{r_1^{d_1}, ..., r_n^{d_n}\}$. In this way, we say that c *enriches* g. Intuitively, a contextual information is a set of resources that can be used to describe a trajectory. A *semantic trajectory* for a segmented trajectory σ_o is a sequence $\tau_o = (< g_1, c_1 >, ..., < g_n, c_n >)$, such that $< g_i, c_i >$ is a pair indicating that g_i is enriched with contextual information c_i.

We also define a particular kind of enriched trajectory, called *labeled trajectories*. Labeled trajectories arise from mobility data captured from social media. We define labeled trajectories as follows. A *labeled trajectory* for a segmented trajectory σ_o is a sequence $\lambda_o = (< g_1, l_1 >, ..., < g_n, l_n >)$, such that $< g_i, l_i >$ is a pair indicating that segment g_i is enriched with a set l_i of labels.

Given a labeled trajectory $\lambda_o \in \Lambda$ of a segmented trajectory σ_o and a set D of available datasets, generate a list $R = [d_1, ..., d_n]$ of datasets such that $d_i \in D$ and d_i likely contains the resources for the semantic enrichment of σ_o. The list should be ranked according to the likelihood that a dataset contains semantic enrichments for σ_o. More formally, let

i. Σ be a set of segmented trajectories
ii. Λ be a set of labeled trajectories of the trajectories in Σ
iii. T be a set of semantic trajectories of the trajectories in Σ
iv. Δ be the set of datasets of the contextual resources of the trajectories in T
v. P be an assessment function that estimates the likelihood that a dataset d_i contains enrichments for $\sigma_o \in \Sigma$ with respect to $\lambda_o \in \Lambda$.

One wants to find a ranking function $rank : \Lambda \mapsto \bigcup_{n=1}^{\infty} \Delta^n$ such that if $rank(\lambda_o) = [d_1, ..., d_n]$ then $P(\lambda_o, d_i) > P(\lambda_o, d_{i+1})$, for $i = 1, ..., n-1$. We segment trajectories with the stop-and-move strategy [11] and label each segment with taxonomic classifications of the place visited at the end of the segment. We cast the problem as a supervised multi-class classification problem. If one takes the set of available datasets as classes of trajectories, one can induce a ranking function as follows. A *classification model* is a function $C : \Lambda \mapsto \bigcup_{n=1}^{\infty} (\Delta \times \Re)^n$ that assigns each labeled trajectory λ_o to a list with n pairs (d, s), where $d \in \Delta$ is a dataset and s is the *assessment score* of d, represented by a Real number. Let $\mathcal{C}$ be the set of all classification models. Let $2^{\Lambda \times T}$ be the set of sets of pairs (λ_o, σ_o). Intuitively, $\Theta \in 2^{\Lambda \times T}$ is a set of pairs (λ_o, σ_o), where λ_o is a labeled trajectory and σ_o is a semantic trajectory, such that the pairs in Θ will be used for training a classification model. Then, we introduce the function $Modeling : 2^{\Lambda \times T} \mapsto \mathcal{C}$ to represent a machine-learning-based process that takes as input sets of pairs of labeled trajectories and that corresponding semantic trajectories, called a *training set*, and outputs a classification model.

Finally, the *ranking function induced by a classification model* C is defined as $rank(\lambda_o) = sortDescending(C(\lambda_o))$, where *sortDescending* sorts pairs by the second coordinate in descending order.

As for the features of trajectories, we tested four types of sets: the set of labels of the places visited in a trajectory (W_{λ_o}), e.g. {`Residence, Law School, Pizza Place`}, boolean model of the set of labels (X_{λ_o}), as used by Information Retrieval (IR) techniques, the set of all valid sequences of labels visited by a trajectory (Y_{λ_o}), e.g. {(`Residence, Law School, Pizza Place`), (`Residence, Pizza Place`), (`Residence, Law School`), ...}, and the boolean model of the sequences of labels (Z_{λ_o}), also as in IR.

3 Dataset Preparation

This section describes the preparation of the dataset used as training data to validate the proposed technique, i.e., the set of labeled and semantic enriched trajectories. We used a set of 9,594,421 geolocated tweets, between June and July 2014 generated in the city of Rio de Janeiro, as trails of movement of people during the FIFA World Cup 2014. A trajectory is defined as the movement of one person between 4:00 AM and 4:00 AM of the next day. There were 912,643 trajectories with 11 samples (tweets) on the average. Each trajectory was segmented using a stop/move heuristic, labeled with place check-ins and semantically enriched with entities from a set of datasets available on the Web.

Labeled Trajectories - Highly dense sampled trajectories are usually segmented using the *speed* and *minimal stop time* criteria. However, low density trajectories, like social media tracks, are not suitable for this kind of segmentation due to the impossibility of correctly computing the speed. We adopted a simpler heuristic for segmenting low density trajectories, yet following the stop-and-move strategy.

The segmentation is based on the intuition that if the time interval between two consecutive tweets is above a given threshold, the user might have moved from one position to another and, therefore, there would be a move segment from the position of the first tweet to the position of the second tweet. On the other hand, if the time interval is short, the user might be stopped or on the move. This last condition is justified because some mobile applications checks-in users automatically. In some cases, it was observed a series of consecutive tweets with short intervals of time and space, giving the idea that the user was on the move. All consecutive tweets considered to be part of the same move segment can be merged into a single segment, while the tweets in the same static position can be merged with the previously identified segment.

Recall from Sect. 2 that the modeling process receives as input a set of labeled trajectories. The trajectories were labeled with the categories of the places visited by users and enriched with the entities from the datasets of Table 1. The places visited by users were captured from Foursquare check-ins made available through tweets. Each Foursquare check-in contains metadata about the place which includes its classification according to the Foursquare taxonomy. This taxonomy is a three-level hierarchy that has, at the highest level, general categories, such as `Food` and `Event`. On the other hand, the lowest level is a very specific classification that contain, for example, the categories `Preschool` and `Private School`. Both levels, however, would lead to a poor discrimination regarding to the class association. The classification model would either over-associate trajectories with classes or discard some associations. Therefore, the trajectories were enriched with the intermediary level of classification, such as `Breakfast Spot`, `Coffee Shop` and `Beach`. The labeling procedure labels segments with information about the place at the end of the segment.

It is important to remark that only trajectories that visited three or more places were selected. It seems not make sense to classify trajectories with small sets of visited places since the induced purpose of the trajectories might be hidden. Therefore, we empirically considered trajectories with three or more places. So, the total number of trajectories was reduced from 912,643 to 8,730.

Semantic Trajectories - Regarding semantic enrichment, we used datasets (Table 1) made available through the open data portals http://dados.gov.br (Portal Brasileiro de Dados Abertos - ODBr) and http://data.rio.rj.gov.br (Portal de Dados abertos da Prefeitura do Rio - ODRio). The enrichment process was semi-automatic and matched the metadata of the places visited by the users (captured from the Foursquare check-ins) with the metadata of entities contained in each dataset.

Table 1. Datasets used for semantic enrichments of the trajectories.

Dataset URI	Source	Alias
http://dados.gov.br/dataset/instituicoes-de-ensino-basico	ODBr	`schools`
http://dados.gov.br/dataset/instituicoes-de-ensino-superior	ODBr	`universities`
http://data.rio.rj.gov.br/dataset/pontos-turisticos-e-culturais	ODRio	`attractions`
http://data.rio.rj.gov.br/dataset/hoteis	ODRio	`hotels`
http://data.rio.rj.gov.br/dataset/museus	ODRio	`museums`
http://data.rio.rj.gov.br/dataset/teatros	ODRio	`theaters`
http://data.rio.rj.gov.br/dataset/estabelecimentos-de-saude	ODRio	`hospitals`
http://data.rio.rj.gov.br/dataset/unidades-administrativas	ODRio	`offices`

The matching process consisted in computing a similarity measure between places p and entities e and manually deciding the matchings. The similarity function used is defined as follows.

$$sim_N(p,e) = 1 - \frac{levenshteinDistance(p[name], e[name])}{p[name].length + e[name].length} \tag{1}$$

$$sim_G(p,e) = \frac{1}{(0.19 \cdot geoDistance(p[position], e[position]) + 1)} \tag{2}$$

$$sim(p, e) = harmonicMean(sim_N(p, e), sim_G(p, e)) \tag{3}$$

The sim_G is defined such that $sim_G = 1$ for $distance = 0$, $sim_G = 0$ for $distance \rightarrow \infty$ and $sim_G = 0.05$ for $distance = 100\,\text{m}$. A place and an entity are matching candidates iff $sim(p, e)$ is greater than 0.95. The matching task only aimed at providing a Gold Standard.

4 Experiments

The main goal of the experiments was to assess the performance of the ranking function. As for the performance measure, the experiments computed the Mean Average Precision (MAP) of the ranking. The next subsections describe the creation of the classification model, the ranking function and the ranking assessment.

Classification Model - We investigated different classification algorithms and concluded that the best classification function is a combination of binary classifiers using the JRip algorithm [1]. Table 2.a compares the F-Measure of different classification algorithms, JRip, J48 [6], OneR [4], ConjunctiveRule [9] and DecisionStump [9], with respect to a positive classification for the classes `offices, theaters, hotels, hospitals, museums, attractions, ies, schools`. This experiment used the set of valid sequences of places (Z_{λ_o}) for the features of trajectories. As we show, none of the algorithms statistically improves the performance of the reference algorithm (JRip). The statistic significance was determined by the paired T-Test method using a set of 10 randomly partitions of the test dataset of the type 2/3 *for training set* + 1/3 *for test set.*

Table 2. F-measure of classification algorithms.

a) Using binary vector of sequences of places.

Dataset	(1)	(2)	(3)	(4)	(5)
`offices`	0.66	0.10	0.00 ●	0.00 ●	0.00 ●
`theaters`	0.82	0.63	0.66	0.66	0.66
`hotels`	0.14	0.03 ●	0.03 ●	0.00	0.00
`hospitals`	0.53	0.32	0.26 ●	0.00 ●	0.00 ●
`museums`	0.58	0.28 ●	0.16	0.00 ●	0.00 ●
`attractions`	0.69	0.69	0.53 ●	0.62	0.53 ●
`universities`	0.82	0.81	0.78	0.73	0.78
`schools`	0.72	0.71	0.71	0.68	0.71
Average	0.62	0.45	0.39	0.34	0.34

b) Multi-class version of JRip binary vector of sequences of places

Class	F-Measure
`offices`	0.31
`theaters`	0.67
`hotels`	0.06
`hospitals`	0.33
`museums`	0.63
`attractions`	0.65
`universities`	0.79
`schools`	0.69
`None`	0.90
Average	0.56

○, ● statistically significant improvement or degradation

(1) rules.JRip '-F 3 -N 2.0 -O 6 -S 1'
(2) trees.J48 '-C 0.25 -M 2'
(3) rules.OneR '-B 6'
(4) rules.ConjunctiveRule '-N 3 -M 2.0 -P -1 -S 1'
(5) trees.DecisionStump ''

The low performance of the classification with respect to the dataset `hotels` can be explained by its generality. That is, a `hotel` can be a stop in several different sequences of places, which makes it more difficult to find a pattern of correlation between trajectories and datasets. The average performance of JRip was 62 %. Table 2.b shows the performance of the multi-class version of JRip, which has an average performance of 56 %. The binary classifiers, therefore, proved to be more efficient. Tables 3.a and b show the performance measures using the set of features in Definitions W_{λ_o} and X_{λ_o}. The best performance was achieved with the JRip algorithm using the categories of the places visited by a trajectory (Definition X_{λ_o}), which was 61 %. These results corroborate the intuition that using sequences of places is more discriminating. For example, a sequence of places (Definition Z_{λ_o}) such as `[Residence, School, Residence]` could indicate that the person is a Student, while a sequence `[Residence, School, School, School, Residence]`, if the schools are different, could indicate that the person is, for example, a professional delivery boy. Both cases, however, have the same set of features. In the first example, the enrichment with a dataset of schools would make sense, while in the last one it seems not to be the case.

Table 3. F-measure of classification algorithms.

a) Using sets of categories as features.

Dataset	(1)	(2)	(3)	(4)
offices	0.17	0.07	0.07	0.00
theaters	0.62	0.53	0.53	0.00 ●
hotels	0.00	0.10	0.10	0.00
hospitals	0.27	0.08 ●	0.08 ●	0.00 ●
museums	0.44	0.27	0.27	0.00 ●
attractions	0.61	0.66	0.66	0.00 ●
universities	0.77	0.62 ●	0.62 ●	0.00 ●
schools	0.71	0.58	0.58	0.00 ●
Average	0.45	0.36	0.36	0

○, ● statistically significant improvement or degradation

b) Using binary vector of categories as features.

Dataset	(1)	(2)	(3)	(4)
offices	0.00	0.00	0.53 ○	0.57 ○
theaters	0.67	0.67	0.86	0.86
hotels	0.26	0.26	0.12	0.14
hospitals	0.28	0.28	0.52 ○	0.45 ○
museums	0.24	0.24	0.72 ○	0.64 ○
attractions	0.68	0.68	0.69	0.74 ○
universities	0.69	0.69	0.83 ○	0.83
schools	0.68	0.63	0.69 ○	0.69
Average	0.43	0.43	0.62	0.62

(1) bayes.NaiveBayesUpdateable ''
(2) bayes.NaiveBayes ''
(3) rules.JRip '-F 9 -N 2.0 -O 6 -S 1'
(4) trees.J48 '-C 0.25 -M 2'

Ranking Assessment - Rankings were generated, as before, using a set of 10 randomly partitions of the enriched dataset such that 2/3 were used for training set and 1/3 for test set. We used sets of binary classifiers, one for each dataset, based on JRip algorithm which is a rule-based classifier that, while trained, generates classification rules such as

> **Rule**: if the a person visited a place of type `States & Municipalities` and did not visit a `Residence` and moved from a place of type `States & Municipalities` to a place of type `Food` then classify trajectory as `attraction`

The training step computes, for each rule, its precision, recall and F-measure. We used the F-measure as an estimate of the confidence of the rule, since intuitively the higher the precision and recall are, the higher the confidence on the classification will be. The confidence, therefore, was used as the *assessment score* (Sect. 2) output by the *classification model.*

To assess the ranking function we computed the Mean Average Precision (MAP) of the rankings of a set of trajectories. We assessed the ranking function on 2,508 trajectories out of the 8,730 trajectories available in the dataset. These trajectories had 1.5 relevant datasets on the average and the computed MAP was 66 %, which means that one would need, on the average, just the three top most entries of the rank to find two datasets for enrichments.

5 Conclusions

This work proposes a novel approach for finding datasets for semantic enrichment based on the types of places visited. The technique takes advantage of place check-ins available on social networks to identify the sequences of places. It is a supervised approach that uses a set of semi-automatically enriched trajectories to learn correlations between the places visited and the datasets available for enrichment. We investigated different classification algorithms and different sets of features for the trajectories. The best performance was obtained with the JRip algorithm and sets of features that contain all possible sequences of places for a trajectory. The resulting ranks obtained, on the average, a MAP of 66 % in the experiments.

Acknowledgements. This work has been funded by CNPq/BR and FAPERJ under grants E-26-170.028/2008, 557128/2009-9, 248743/2013-9, 248987/2013-5, 303332/2013-9, 442338/2014-7, 444976/2014-0 and E-26-201.337/2014.

References

1. Cohen, W.W.: Fast effective rule induction. In: the 12th International Conference on Machine Learning, pp. 115–123. Morgan Kaufmann (1995)
2. Fileto, R., Krüger, M., Pelekis, N., Theodoridis, Y., Renso, C.: Baquara: a holistic ontological framework for movement analysis using linked data. In: Ng, W., Storey, V.C., Trujillo, J.C. (eds.) ER 2013. LNCS, vol. 8217, pp. 342–355. Springer, Heidelberg (2013)
3. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space, vol. 1. Morgan & Claypool, San Rafael (2011)
4. Holte, R.C.: Very simple classification rules perform well on most commonly used datasets. Mach. Learn. **11**(1), 63–91 (1993)
5. Nabo, R.G.B., Fileto, R., Nanni, M., Renso, C.: Annotating trajectories by fusing them with social media users posts. In: the XI Brazilian Symposium on GeoInformatics, pp. 25–36 (2014)
6. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, Burlington (1993)

7. Spinsanti, L., Celli, F., Renso, C.: Where you stop is who you are: understanding peoples activities. In: the 5th Workshop on Behaviour Monitoring and Interpretation–User Modelling (2010)
8. Uzun, A.: Linked crowdsourced data - enabling location analytics in the linking open data cloud. In: 2015 IEEE International Conference on Semantic Computing, pp. 40–48. IEEE (2015)
9. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Burlington (2011)
10. Yan, Z., Chakraborty, D., Parent, C., Spaccapietra, S., Aberer, K.: SeMiTri: a framework for semantic annotation of heterogeneous trajectories. In: The 14th International Conference on Extending Database Technology, pp. 259–270 (2011)
11. Yan, Z., Chakraborty, D., Parent, C., Spaccapietra, S., Aberer, K.: Semantic trajectories: mobility data computation and annotation. Trans. Intell. Syst. Technol. **4**(3), 49 (2013)
12. Yuan, J., Liu, X., Zhang, R., Sun, H., Guo, X., Wang, Y.: Discovering semantic mobility pattern from check-in data. In: Benatallah, B., Bestavros, A., Manolopoulos, Y., Vakali, A., Zhang, Y. (eds.) WISE 2014. LNCS, vol. 8786, pp. 464–479. Springer, Heidelberg (2014). doi:10.1007/978-3-319-11749-2_35

On Impact of Weather on Human Mobility in Cities

Jun Pang(✉), Polina Zablotskaia, and Yang Zhang

Computer Science and Communications, University of Luxembourg,
Luxembourg, Luxembourg
jun.pang@uni.lu

Abstract. Although researchers have demonstrated that human mobility is constrained by space, time and social relations, one important factor, namely weather, has been often ignored in the literature. Not only influences what people wear everyday, weather also has a major impact on their mobility. In this paper, we conduct the first large-scale analysis of weather's impact on human mobility in cities. Focusing on a number of major cities, we construct a human mobility dataset from the social network Instagram. We discover that in general nice weather (e.g., moderate temperature and high pressure) has a positive impact on human mobility. Through analyzing mobility at locations of different categories, we further discover that human mobility is less influenced by weather at certain categories such as residences than others including stores and entertainment places.

1 Introduction

Urbanization is a massive process happening in this century. Every year, more and more people are moving to cities. According to a UN report[1], by 2050 more than 6 billion people will live in cities. Although living in cities brings a lot of convenience to people, it also causes major problems, such as air pollution and traffic congestion. While much effort has been taken to tackle these problems, one fundamental challenge is to fully understand how people move, i.e., human mobility in cities.

Human mobility has attracted the research community a considerable amount of interest during the past decade. Researchers have demonstrated that human mobility is constrained by space, time [1] and social networks [2]. On the other hand, another important factor, i.e., weather, receives much less attention and is often ignored. Weather as a natural phenomenon influences our mobility in many ways. When people check weather reports, they not only decide what to wear, but also where to visit. For instance, few people are willing to walk in a park on a cloudy winter afternoon.

Understanding the relationship between mobility and weather can result in positive benefits for multiple stakeholders: for example, city governors can design

[1] http://bit.ly/1N3gAH6.

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 247–256, 2016.
DOI: 10.1007/978-3-319-48743-4_20

Table 1. Dataset summary.

City	#Check-ins	#Users	#Locations	City	#Check-ins	#Users	#Locations
New York	2,728,705	788,980	30,644	Washington DC	542,822	185,687	10,601
Los Angeles	2,011,106	607,380	27,716	San Francisco	635,842	225,438	9,620
Tokyo	891,029	300,111	26,586	Chicago	725,223	233,844	12,407
London	1,441,658	516,640	15,571	Rome	232,305	102,022	6,267
Paris	633,868	253,516	11,112	Milan	296,353	122,481	5,917
Boston	465,615	165,166	7,619	Barcelona	245,298	113,997	5,457
Hongkong	191,899	87,413	4,203				

specific plans for different weather conditions to control traffic flow; shop owners can provide suitable benefits to attract customers; city residents can choose to visit less crowded places on weekends. In the current work, we conduct the first large-scale analysis of weather's impact on human mobility in cities. Our contributions can be summarized as follows:

- We construct a mobility dataset under different weather parameters for 13 major cities across the world (Sect. 2). We gather more than 10 millions of users' location records, namely check-ins, from Instagram and weather data from Forecast.io.
- We analyze the relationship between users' general mobility behaviors and different weather parameters such as temperature and humidity (Sect. 3). We quantify users' mobility through average check-in volumes, average movement volumes and average movement distances. Our discoveries, for example, include both low and high temperature have negative effects on mobility; high pressure on the other hand positively affects mobility. Interestingly, we also discover that humidity affects mobility negatively in coastal cities while positively in inland cities.
- We take one step further to analyze users' mobility at different location categories under different weather parameters (Sect. 4). We discover that users' average check-in volumes at locations of certain categories, such as store, entertainment and professional places, are more correlated with weather than others such as residence places. Moreover, people' movement patterns among location categories are less diverse under a uncomfortable weather condition than a comfortable one.

2 Dataset Construction

Check-in Data. We collect the geo-tagged photos, i.e., check-ins, in 13 major cities worldwide from Instagram by using its public API from August 1st, 2015 until December 15th, 2015. We first resort to Foursquare to extract all location ids within each city we are interested in, meanwhile collect each location's category information. Then for each Foursquare's location id, we query Instagram's API to get its corresponding location id in Instagram. After obtaining

Instagram's location ids, we query each location's recent check-ins several times a day. In the end, more than 10M check-ins have been collected. Table 1 summarizes the dataset. As Foursquare organizes location categories into a tree structure, we take its first level categories including *entertainment*, *food*, *bar*, *outdoor*, *professional*, *residence*, *store* and *transportation* to label each location.

Weather. We exploit Forecast.io's API to extract weather data. Forecast.io is a weather application started in 2013, it gathers the data from multiple sources such as NOAA and Met Office, and provides users with the aggregated results. Forecast.io's API provides daily weather data covering temperature (°C), humidity (relative humidity), wind speed (miles per hour), pressure (millibar). In addition, as people normally do not feel the difference when the temperature varies one or two degrees, we bucket temperature into bins of 3° starting from 0 °C (−2 °C–0 °C) to 30 °C (28 °C–30 °C).

3 Weather and Mobility

For weather, four parameters are considered including temperature, pressure, wind speed and humidity. For mobility, we focus on two aspects. The first one is the average number of check-ins, namely *average check-in volumes*, under each value of each weather parameter. The second one is related to users' movements, including the average number of movements, namely *average movement volumes* and *average movement distances* (km). Here, we consider a user checking in at two locations within a certain time threshold τ as one movement. In this paper, we choose τ to be 3 hours which we believe is a reasonable transition time for a user. The same formulation has been used in [3] as well. Note that our mobility quantification is at a general level instead for each individual user, i.e., we calculate the mean of users' total number of check-ins, number of movements and movement distances under each value of each weather parameter.

Temperature. Figure 1 depicts the average check-in volumes under different temperature (bucketed by 3°C) in Paris and Boston. We observe that people check in more often under a moderate temperature than low (≤6 °C) and high (≥24 °C) temperature in both cities. We further fit the data into a Gaussian

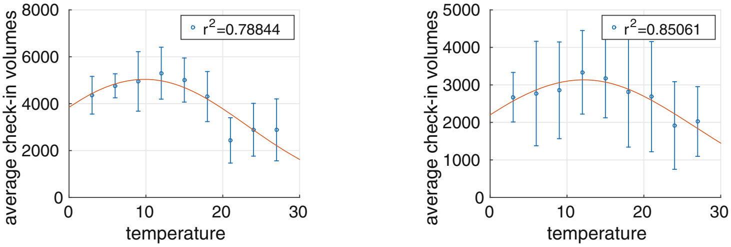


Fig. 1. Temperature vs. average check-in volumes in Paris (left) and Boston (right).

Table 2. r^2 Between temperature and average check-in volumes.

City	r^2	b	c	City	r^2	b	c
New York	0.90	12.76	25.72	Washington DC	0.84	14.45	19.21
Los Angeles	0.36	17.97	17.78	San Francisco	0.86	12.52	17.21
Tokyo	0.67	13.77	22.92	Chicago	0.45	13.64	23.62
London	0.24	10.97	18.92	Rome	0.85	14.71	15.78
Paris	0.78	9.83	14.39	Milan	0.91	10.52	15.36
Boston	0.85	12.10	20.37	Barcelona	0.64	16.82	21.11
Hongkong	0.77	22.22	21.14				

Table 3. r between pressure and mobility ($r1$: average check-in volumes, $r2$: average movement volumes, $r3$: average movement distances).

City	$r1$	$r2$	$r3$	City	$r1$	$r2$	$r3$
New York	0.23	0.60	0.68	Washington DC	0.74	0.69	0.66
Los Angeles	0.16	0.21	0.12	San Francisco	0.31	-0.20	-0.22
Tokyo	0.81	0.64	0.55	Chicago	0.30	0.35	0.39
London	0.63	0.60	0.63	Rome	0.57	0.09	0.05
Paris	0.54	0.48	0.47	Milan	0.48	0.18	0.18
Boston	-0.05	0.56	0.59	Barcelona	-0.17	-0.43	-0.53
Hongkong	0.25	0.18	0.18				

function defined as $avg\_ci(t) = a \cdot \exp(-(\frac{t-b}{c})^2)$. In the formula, $avg\_ci(t)$ is the average check-in volume at temperature t (bucketed by 3°C), a, b and c are the parameters of the function: a represents the height of the curve peak, b marks the center of the curve and c controls the width. As shown in Fig. 1, a high coefficient of determination, i.e., r^2, is obtained for the fitting, meaning that a Gaussian curve captures the relation between temperature and average check-in volumes. Table 2 lists r^2 together with parameters b and c for all the cities. We make two interesting observations. First, data for most cities fit the Gaussian function well, except for Los Angeles and London with relatively weak results. This indicates that there exists a universal pattern of temperature's impact on human mobility. Second, the central point of the Gaussian function, i.e., b, varies across the cities. Cities located in hot regions such as Hongkong and Los Angeles have higher values for b, as people living there are used to hot weather, compared to cities located in cold regions, e.g., London.

For the second aspect of mobility, i.e., average movement volumes and average movement distances, as an example we plot the average movement volumes in Rome and the average movement distances in Washington DC as a function of temperature in Fig. 2, respectively. Consistently, we see – similar to average check-in volumes – both average movement volumes and distances fit Gaussian functions well.

From the above analysis, we first conclude that human mobility is more adapted to moderate temperature than both low and high temperature.

Pressure. High pressure is a whirling mass of cool and dry air which generally brings good weather, while low pressure is normally associated with bad weather such as cloud, rain and wind. We expect that pressure has positive effects on users' mobility.

Table 3 lists three correlation coefficients (r) between pressure and mobility, pressure indeed positively affects users' mobility in most of the cities. Especially for Tokyo and London, we observe strong correlations (see Fig. 3). On the other hand, Barcelona is the only city with three negative correlation coefficients, indicating that pressure has negative effects on human mobility in Barcelona. In addition, most of the cities show consistency between average check-in volumes ($r1$) and movements ($r2$, $r3$), except for San Francisco and Boston. People in San

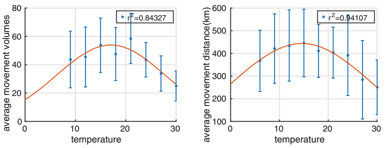


Fig. 2. Temperature vs. average movement volumes in Rome (left) and average movement distances in Washington DC (right).

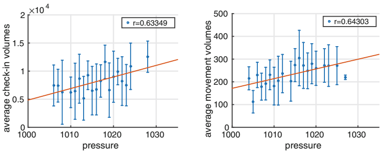


Fig. 3. Pressure vs. check-in volumes in London (left) and movement volumes in Tokyo (right).

Francisco have more check-ins but less and shorter movements on high pressure days. In Boston, high pressure does not affect average check-in volumes much, instead it leads to more and longer movements.

Wind Speed. Wind speed is an important aspect of weather. Through analysis, we discover that average check-in volumes in most cities receives negative wind speed effects (Table 4). Figure 4 (left) presents the result in Barcelona as an example.

The relation between wind speed and movements, on the other hand, is more complicated. For some cities including Los Angeles (see Fig. 4 (right)), Tokyo, Paris, Washington DC and Hongkong, wind speed has similar effects on movements as on check-in volumes. On the other hand, in New York, Boston, San Francisco and Barcelona, the (negative) effects of wind on movements become weaker. Moreover, in Chicago, Rome and Milan, there exist positive effects of wind speed on movements. One explanation could be wind negatively affects cycling and walking which results in more car and public transportation usage in these cities. In turn, this leads to the increases in movement volumes and distances. In the end, we observe that in London wind speed has weak effects on average check-in volumes but strong (negative) effects on movements.

Humidity. People normally feel uncomfortable when humidity is low (≤ 0.3) or high (≥ 0.8). Therefore, similar to the case of temperature, we expect the relation between mobility and humidity to follow a Gaussian curve as well. However, analysis results show that humidity (mostly between 0.3 and 0.8) and mobility are linearly correlated in most of the cities. More interestingly, we observe contradictory linear correlations in different cities. As shown in Fig. 5, humidity has positive effects on mobility in Rome while negative effects in Hongkong. Table 5 lists correlation coefficients for both kinds of cities (the results for movements are quite similar and omitted).

Fully understanding the correlation between humidity and mobility involves multiple factors, such as city location, temperature or even culture background, which is out of the scope of the current work. On the other hand, by only studying the dataset, we observe that most of the cities with positive humidity effects are inland cities except for Barcelona and Boston. On the other hand, cities with negative effects are all coastal cities where humidity is normally high.

Table 4. r between wind and mobility ($r1$: average check-in volumes, $r2$: average movement volumes, $r3$: average movement distances).

City	$r1$	$r2$	$r3$	City	$r1$	$r2$	$r3$
New York	−0.47	−0.05	−0.07	Washington DC	−0.29	−0.25	−0.24
Los Angeles	−0.60	−0.60	−0.55	San Francisco	−0.50	−0.16	−0.16
Tokyo	−0.37	−0.40	−0.31	Chicago	−0.26	0.26	0.30
London	0.02	−0.57	−0.56	Rome	−0.05	0.33	0.45
Paris	−0.16	−0.28	−0.36	Milan	−0.31	0.24	0.21
Boston	−0.40	−0.03	−0.01	Barcelona	−0.56	−0.18	-0.07
Hongkong	0.07	0.12	0.02				

Table 5. r between humidity and average check-in volumes.

City	r	City	r
Hongkong	−0.70	Rome	0.68
Los Angeles	−0.54	Paris	0.61
New York	−0.28	London	0.59
Tokyo	−0.19	Milan	0.56
San Francisco	−0.14	Barcelona	0.50
		Boston	0.33
		Chicago	0.16
		Washington DC	0.12

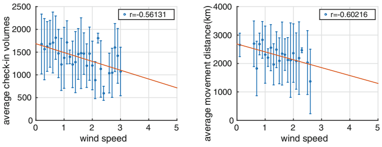


Fig. 4. Wind vs. average check-in volumes in Barcelona (left) and average movement distances in Los Angeles (right).

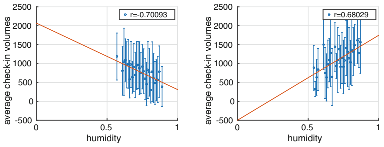


Fig. 5. Humidity vs. average check-in volumes in Hongkong (left) and Rome (right).

We conjecture that humidity negatively affects human mobility in coastal cities while positively in inland cities.

4 Weather and Location Category

In this section we take one step further to analyze weather's influences on mobility at different location categories. We start by analyzing average check-in volumes at each category, then discuss movement patterns among categories.

4.1 Average Check-In Volumes

Temperature. We exploit Gaussian function to model the relation between temperature and average check-in volumes at each location category, assuming that users are more adapted to moderate temperature than both low and high temperature (Sect. 3). Through analysis, we obtain high correlation of determination (r^2) for Gaussian function fitting at entertainment, professional, outdoor and store places. On the other hand, residence is the category with the lowest r^2 values, followed by food and bar. Figure 6 (left) further plots the results in Los Angeles, London and Chicago.

Since Gaussian function cannot capture the correlation between temperature and average check-in volumes at residence, food and bar places, we further examine the data of these categories. Figure 7 depicts the average check-in volumes at

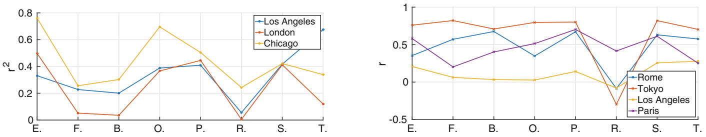


Fig. 6. r^2 of fitting between average check-in volumes and temperature at location categories (left), r between average check-in volumes and pressure at different location categories (right) (each category is denoted by its first letter).

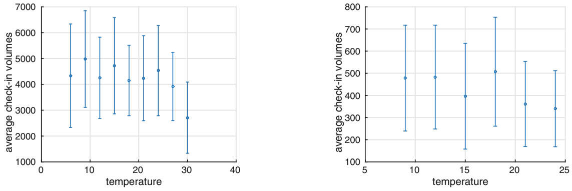


Fig. 7. Average check-in volumes at food places in New York (left) and bar places in San Francisco (right) under different temperature.

food places in New York and bar places in San Francisco as a function of temperature; no clear correlation can be observed. This means whether people going to a bar (a restaurant or a residence) is not strongly dependent on temperature. One reason could be that places of these categories are mostly indoor places, thus not weather-exposed. We conclude that human mobility is more affected by temperature at entertainment, professional, outdoor and store places while it is less affected by temperature at food, bar and residence places.

Pressure. Section 3 states that pressure positively affects users' mobility, this result holds for most of the location categories as well. In addition, in most cities, users' mobility at entertainment, professional and store places receives more positive pressure effects than mobility at other categories. On the other hand, the correlation at residence places is rather weak. For instance, in Fig. 6 (right), correlation coefficients (r) at residence places in Tokyo, Rome, Los Angeles and Paris drop quickly when compared to other categories. Meanwhile, there also exist subtle differences among the cities. For instance, users' average check-in volumes at food places have the highest pressure effects in Tokyo while food places have the lowest pressure effects for Paris.

Wind Speed. Previously, we have shown that even though wind speed has different effects on movements (average movement volumes and distances) in different cities (Table 4), it still negatively affects average check-in volumes in most cities. However, when conducting analysis at the location category level, similar pattern between wind speed and average check-in volumes cannot be observed. To give an example, we discover that the impact of wind speed at residence places decreases (similar to the cases of temperature and pressure) in some cities, while in other cities the wind's negative impact even gets stronger.

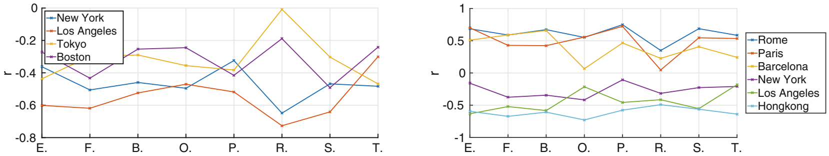


Fig. 8. r between average check-in volumes and wind speed (left) humidity (right) at different location categories.

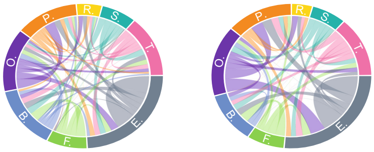


Fig. 9. Movements among categories in London at 12 °C (left) and 24 °C (right).

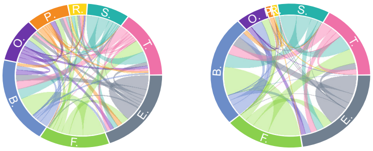


Fig. 10. Movements among categories in Los Angeles at humidity 0.56 (left) and 0.8 (right).

From Fig. 8 (left), we see that Tokyo and Boston belong to the former case while New York and Los Angeles represent the latter one.

Humidity. Effects of humidity on users' mobility are positive in some cities while negative in others. For most cities with positive humidity effects, residence is again the category with the lowest correlation coefficients (r) while entertainment, professional and store places have the highest (similar to the results of pressure). For cities with negative humidity effects, we cannot observe a clear pattern. Figure 8 (right) plots the results of a few cites: Rome, Paris and Barcelona for the first case, New York, Los Angles and Hongkong for the second.

4.2 Movements Among Location Categories

Since each location is associated with a location category, we further study the weather's impact on movements among different categories. In the current work, we focus on the direct movements between two categories, e.g., from professional places to food places.

Temperature. Among all the 13 cities, we discover that users' movements among location categories under a moderate temperature are more diverse than those under low and high ones. Figure 9 plots two chord diagrams in London under a moderate temperature (12 °C) and a high temperature (24 °C). Each location category is represented by its first letter (capitalized) on the circle, links having the same color as a category are movements starting from that category. Width of each link is proportional to the number of movements. In Fig. 9, there exist more links among categories when temperature is 12 °C than 24 °C. For example, there are many links from entertainment to outdoor and transportation places in the left part of Fig. 9, while the number of links decreases on the right part of Fig. 9; the most likely destinations for people after checking in at transportation places are bar places when temperature is 12 °C, while they are professional places when temperature is 24 °C.

Humidity. Similar to the case of temperature, users' movements among location categories are more diverse under a comfortable humidity condition than a uncomfortable one. Figure 10 plots the chord diagrams in Los Angeles under two humidity conditions, i.e., 0.56 and 0.8. There are more links in Fig. 10 with humidity condition 0.56 than with 0.8, e.g., many more movements end at outdoor places in the left chord diagram than in the right one. Similar results are obtained for pressure as well.

5 Related Work

To the best of our knowledge, the current work is the first large-scale study on weather and human mobility. One close line of work is the study on weather and transportation carried out by the transportation community [4].

Comparing to these studies, our work has the following advantages. (1) Most of the studies conducted by the transportation community focus on weather's impact on people's transportation modes such as public transportation, bicycle or walk. Especially, bicycle usage attracts a lot of attentions (e.g., see [5,6]). On the other hand, we focus on users' mobility without any constrains, this makes our analysis more general than theirs. (2) Our dataset is at the global level, i.e., we focus on the mobility of users among 13 cities located in Asia, Europe and North America, while most of the datasets used by the transportation community concentrate on a single city or country. Besides, since our mobility data is from Instagram, the user sample is much bigger than those works whose data is normally collected by conducting surveys.

6 Conclusion

We have conducted the first large-scale analysis on the relationship between weather and human mobility in cities. Our discoveries include (1) nice weather, characterized by moderate temperature, high pressure, slow wind speed and suitable humidity, has positive effects on users' mobility; (2) users' mobility at certain location categories, e.g., residence places, is less influenced by weather than mobility at other categories including entertainment, professional and store places.

References

1. Gonzalez, M., Hidalgo, C., Barabasi, A.L.: Understanding individual human mobility patterns. Nature **453**, 779–782 (2008)
2. Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: Proceedings of 17th ACM Conference on Knowledge Discovery and Data Mining (KDD), pp. 1082–1090. ACM (2011)

3. Noulas, A., Shaw, B., Lambiotte, R., Mascolo, C.: Topological properties and temporal dynamics of place networks in urban environments. In: Proceedings of 24th International Conference on World Wide Web (WWW Companion), pp. 431–441. ACM (2015)
4. Böcker, L., Dijst, M., Prillwitz, J.: Impact of everyday weather on individual daily travel behaviours in perspective: a literature review. Transp. Rev. **33**(1), 71–91 (2013)
5. Hanson, S., Hanson, P.: Evaluating the impact of weather on bicycle use. Transp. Res. Rec. **629**, 43–48 (1977)
6. Nankervis, M.: The effect of weather and climate on bicycle commuting. Transp. Res. Part A: Policy Pract. **33**(6), 417–431 (1999)

Graph Theory

Minimum Spanning Tree on Uncertain Graphs

Anzhen Zhang(✉), Zhaonian Zou, Jianzhong Li, and Hong Gao

Harbin Institute of Technology, Harbin 150001, China
azzhang@hit.edu.cn

Abstract. In recent years, lots of data in various domain can be represented and described by uncertain graph model, such as protein interaction networks, social networks, wireless sensor networks, etc. This paper investigates the most reliable minimum spanning tree problem, which aims to find the minimum spanning tree (MST) with largest probability among all possible MSTs on uncertain graphs. In fact, the most reliable MST is an optimal choice between stability and cost. Therefore it has wide applications in practice, for example, it can serve as the basic constructs in a telecommunication network, the link of which can be unreliable and may fail with certain probability. A brute-force method needs to enumerate all possible MSTs and the time consumption grows exponentially with edge size. Hence we put forward an approximate algorithm in $O(d^2|V|^2)$, where d is the largest vertex degree and $|V|$ is vertex size. We point out that the algorithm can achieve exact solution with expected probability at least $(1-(\frac{1}{2})^{(d+1)/2})^{|V|-1}$ and the expected approximation ratio is at least $(\frac{1}{2})^{d|V|}$ when edge probability is uniformly distributed. Our extensive experimental results show that our proposed algorithm is both efficient and effective.

1 Introduction

Recently, lots of data in various domain can be represented by graph model, such as the web, social networks, and cellular systems. Such networks are often subject to uncertainties caused by noise, incompleteness and inaccuracy in practice [1]. Incorporating uncertainty to graphs leads to uncertain graphs, each edge of which is associated with an edge existence probability to quantify the likelihood that this edge exists in the graph. For example, in a telecommunication network, a link can be unreliable and may fail with certain probability [14]; in a social network, the probability of an edge may represent the uncertainty of a *link prediction* [6]; in a wireless sensor network, communication links between sensor nodes often suffer from inevitable physical interference [15]. The uncertain graph, also referred as probabilistic graph, addresses such scenarios conveniently in a unified way.

There has been extensive research on the minimum spanning tree on exact graphs which are precise and complete. Given a connected exact graph $G = (V, E)$, each edge has a non-negative weight, a spanning tree T of G is a tree whose edges connect all the nodes in G, the sum weight of all edges in T is the cost

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 259–274, 2016.
DOI: 10.1007/978-3-319-48743-4_21

of T. Among all spanning trees the one with minimum cost is minimum spanning tree, which is short for MST. A number of algorithms have been proposed to compute MST for exact graph, among which Prim and Kruskal gave two classical algorithms in polynomial time [2,3].

The problem of computing MST is fundamental for uncertain graphs, just as they are for exact graphs. Obviously we can still obtain the MST with minimum cost in uncertain graph by neglecting the possibility of edges and treat the uncertain graph as exact graph. We call the obtained MST *minimum cost MST*, however, it suffers from low reliability and may fail to exist since we leave the possibility of edges out of consideration. Actually we can obtain the MST with largest probability by setting the weight of edge e to be $-log(p(e))$ and executing MST algorithm for exact graphs. The MST obtained this way has the largest existing probability but has no guarantee for the cost, we call it *maximum probability MST* for simplicity. To make a balance between those two types of MST, we propose *the most reliable MST* which has largest probability among all possible MSTs and it can be found in a wide range of network applications.

In a telecommunication network, the most reliable MST can serve as the basic constructs to help connect all nodes in the network at least cost while the stability of network can still be guaranteed [16]. In the scenario of advertisement promotion in social network with edge uncertainty measures the intimacy between two friends and edge weight quantifies the cost of spreading, the most reliable MST can help discover a most effective propagation path with minimum advertisement cost. Another practical application is data aggregation tree in wireless sensor network [15]. Most reliable MST can serve as an optimal initial data aggregation tree with sink node to be the root and source nodes to be leafs since it takes transmission energy cost and link failure probability into consideration, which contributes to the construction of data aggregation tree that maximizes the network lifetime.

In uncertain graph, possible instantiations of the graph are commonly referred to as worlds or implicated graph, the probability of a world is calculated based on the probability of its edges, as will shown in later section. There exists at least one MST in each connective implicated graph, all MSTs of the connective implicated graphs form the MST set of the uncertain graph and the most frequent MST in the set is the most reliable one, denoted by MST_{max}.

A brute-force method is to enumerate all connected implicated graphs and compute MST for each of them, however, enumerating all implicated graph needs $O(2^{|E|})$time, where $|E|$ is the edge size. The exact algorithm is not feasible since the scale of graph is large in practical applications. Hence we present an approximate algorithm in $O(d^2|V|^2)$, where d is the maximum degree of vertexes and $|V|$ is vertex size. Our theoretical analysis shows that it has pretty good performance in aspects of accuracy and approximate ratio. Extensive experiment results on synthetic sets show that our approximate algorithm outperforms the other three algorithms in terms of stability and cost.

The rest of the paper is organized as follows. We define the most reliable MST problem in Sect. 2. An approximate algorithm and its performance analysis is presented in Sect. 3. We show experimental studies in Sect. 4 and present related work in Sect. 5. Finally, we conclude this paper in Sect. 6.

2 Problem Formulation

In this section, we formally present the uncertain graph model and introduce the problem of most reliable MST in an uncertain graph.

2.1 Model of Uncertain Graphs

Let $\mathcal{G} = (V, E, P, W)$ be an uncertain graph, where V and E denote the set of vertexes and edges respectively. $P : E \to (0, 1]$ is a function assigning existence possibility values to edges. W is weight function, and $w(e)$ is the weight of edge $e \in E$.

An uncertain graph has many existence forms due to the uncertainty of edges, each deterministic form $g = (V, E_g)$ is called *implicated graph* and is denoted by $\mathcal{G} \Rightarrow g$. Each edge $e \in E_{\mathcal{G}}$ is selected to be an edge of g with probability $P(e)$. The total number of implicated graphs is $2^{|E|}$ since each edge has two cases as to whether or not that edge is present in the graph. We assume that all existence possibilities of edges are independent, the probability of an uncertain graph $\mathcal{G}$ implicating an exact graph g is

$$P(\mathcal{G} \Rightarrow g) = \prod_{e \in E(g)} P(e) \prod_{e' \in E(G) \setminus E(g)} (1 - P(e')) \tag{1}$$

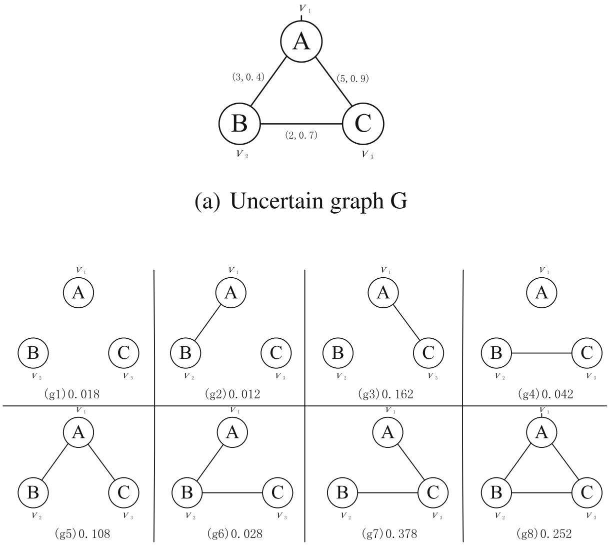
where $P(e)$ is the existence possibility of edge e. To gain more intuition on the uncertain graph model and the implicated graph, We present a simple example in the following.

Example 1. Consider the uncertain graph $\mathcal{G}$ in Fig. 1(a). There are $2^3 = 8$ implicated graphs, and the probability of each graph is calculated based on Eq. (1), as shown in Fig. 1(b). Take $g2$ for example, $P(\mathcal{G} \Rightarrow g2) = 0.4 \times (1-0.9) \times (1-0.7) = 0.012$. The probability of the other implicated graphs are calculated in the same way.

2.2 Most Reliable MST

There exists at least one MST in each connective implicated graph of uncertain graph $\mathcal{G}$ with probability same as the implicated graph containing it, all possible MSTs form MST set $\{MST_1, MST_2, \cdots\}$, and MST_i denotes the edge set of the ith MST.

Obviously a MST for an implicated graph may be MST for another implicated graph, hence each MST in MST set is accompanied with certain frequency and the one with maximum frequency is most reliable, that is MST_{max}. The existing probability $P(MST_i)$ of MST_i is determined by the implicated graph it belongs



(a) Uncertain graph G

(b) Imp(G)

Fig. 1. A simple example

to. As we mentioned above, MST_i could be MST of several implicated graphs, so $P(MST_i)$ is the sum probability of all implicated graphs whose MST is MST_i and can be mathematically quantified as follows.

$$P(MST_i) = \sum_{g \in Imp(G)} P(G \Rightarrow g) \cdot I_1(g) \cdot I_2(g) \tag{2}$$

where $I_1(g)$ and $I_2(g)$ are indicator functions.

$$I_1(g) = \begin{cases} 1 & \text{if } g \text{ is connected} \\ 0 & \text{otherwise} \end{cases}$$

$$I_2(g) = \begin{cases} 1 & \text{if } MST_i \text{ is MST of} g \\ 0 & \text{otherwise} \end{cases}$$

Now we give the formal definition of MST_{max}, that is the MST with maximum probability.

$$MST_{max} = argmax\{P(MST_i)\} \tag{3}$$

In the following example, we show how to exactly compute MST_{max} of uncertain graph $\mathcal{G}$ in Fig. 1(a).

Example 2. There are 8 implicated graphs of $\mathcal{G}$ as shown in Fig. 1(b), only 4 of them are connected, namely, $g5$, $g6$, $g7$ and $g8$. After computing the MST of

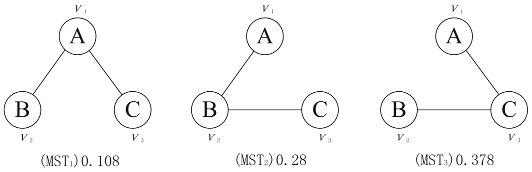


Fig. 2. MST set of graph G

the four subgraphs respectively, we get three different MSTs, MST_1, MST_2 and MST_3 as shown in Fig. 2. The MSTs of $g6$ and $g8$ are actually same to each other, that is MST_2. We can compute the existing probability of MST_i using Eq. (3), $i \in 1, 2, 3$.

$$P(MST_1) = P(\mathcal{G} \Rightarrow g5) = 0.108$$
$$P(MST_2) = P(\mathcal{G} \Rightarrow g6) + P(\mathcal{G} \Rightarrow g8) = 0.028 + 0.252 = 0.28$$
$$P(MST_3) = P(\mathcal{G} \Rightarrow g7) = 0.378.$$

According to Eq. (3), MST_{max} is the MST with largest probability in MST set, which refers to MST_3 in our example, thus we obtain the most reliable MST $MST_{max} = \{AC, BC\}$, and $P(MST_{max}) = P(MST_3) = 0.378$. We denote the cost of MST as $|MST|$, then $|MST_{max}| = |MST_3| = 5 + 2 = 7$.

An intuitive way to compute MST_{max} is to enumerate all implicated subgraphs and compute corresponding MSTs of the connected subgraphs, each MST is with a corresponding existing probability and the one with largest probability is the MST_{max}. Detailed algorithm is shown in Algorithm 1.

Algorithm 1. Naive Algorithm

```
Input: Uncertain graph $\mathcal{G} = (V, E, P, W)$
Output: The edge set $A$ of $MST_{max}$

$MST\_Edge = \varnothing; MST\_P = 0; MST\_W = 0; Imp = \varnothing;$
$map < Edge, (P, W) > MST\_SET$
for $i \leftarrow 2^{|V|-1} - 1$ to $2^{|E|} - 1$ do
    Transform $i$ to its binary form, denoted by $i\_binary$ // i.e.100111
    if The number of 1's in $i\_binary$ is not smaller than $|V| - 1$ then
        Add all edges whose index corresponds to 1 in $i\_binary$ in $Imp$
        $MST\_P = \prod_{e \in Imp} P(e) \prod_{e' \in E \setminus Imp} (1 - P(e'))$
        $MST\_Edge, Imp\_W$ = Prim($Imp$)
        if $MST\_Edge$ is not in $MST\_SET$ then
            Add $(MST\_Edge, (MST\_P, MST\_W))$ in $MST\_SET$
        else if $MST\_Edge$ is in $MST\_SET$ then
            Modify $P = MST\_P \cdot P$ in original $< Edge, (P, W) >$ pair whose $Edge =$
            $MST\_Edge$
return $Edge$ with maximum probability in $MST\_SET$
```

In Naive Algorithm, we enumerate all implicated algorithms and compute MST using Prim algorithm. To find the MST with largest frequency, we use a map whose key is edge set of MST, and value is a pair of probability and weight, the map can hash all MSTs with same edge set together. After computing all MSTs of connected implicated graphs, we find the one with largest probability, that is MST_{max}. There are total $2^{|E|}$ implicated graph of $\mathcal{G}$, the efficiency of the brute-force algorithm is unsatisfactory when applied to large-scale graphs due to its exponential growing rate.

2.3 Edge Induced Combinational Method

According to the definition of most reliable MST in Sect. 2.2, it is not easy to find the most reliable MST in polynomial time. However, we find out another way to compute the probability of MST in a combinational way. Without lose of generality, we suppose that the edge weight is different from the others. For MST_i in MST set, the set of all edges not in MST_i is denoted by R_i, $R_i = E - MST_i$ by definition. For an edge e_i in R_i, adding e_i in MST_i will form a circle due to the connectivity of MST_i. Next we give the definition of safe edge and dangerous edge in R_i.

Definition 1. *Safe edge. For an edge e_i in R_i, it is a safe edge to MST_i if it has largest weight in the circle when adding e_i in MST_i.*

Definition 2. *Dangerous edge. For an edge e_i in R_i, it is a dangerous edge to MST_i if it does not has largest weight in the circle when adding e_i in MST_i, in other words, there exists an edge whose weight is larger than e_i's.*

Based on the above definition, we divide the remaining edge set R_i into two separate sets, namely, RS_i and RD_i. All safe edges in R_i are placed in RS_i, and all dangerous edges are in RD_i. Now we move on to give a combinational way to compute the probability of MST_i and we prove that the probability computed in this way is same as that given by Eq. 2.

$$P(MST_i) = \prod_{e_i \in MST_i} p(e_i) \cdot \prod_{e_j \in RD_i} (1 - P(e_j)) \tag{4}$$

Theorem 1. *The probability calculated by Eq. 4 is equal to that obtained by Eq. 2.*

Proof. We only give a sketch of the proof, the detailed proof is omitted due to page limit. Apparently, the MST_i itself is a implicated graph whose MST is MST_i, based on this we can further add any safe edge to MST_i and they will not affect the MST of newly constructed implicated graphs, that is MST_i. This is because safe edge has largest weight in the circle and will be discarded. Further more, we can prove that any dangerous edge added to MST_i will affect the original structure, therefore all edges in RD_i can not exist, which is shown in Eq. 4.

Example 3. BC is a dangerous edge to MST_1, according to Eq. 4, we have $P(MST_1) = P(AB) \cdot P(AC) \cdot (1 - P(BC)) = 0.4 \times 0.9 \times (1 - 0.7) = 0.108$, which is same as the result in Example 2. $P(MST_2)$ and $P(MST_3)$ can be computed in a similar way.

3 Greedy Algorithm

In this section, we present an approximate algorithm of the most reliable MST. The detailed algorithm description is shown in Sect. 3.1 and we analyse the performance of our approximate algorithm in Sect. 3.2. The complexity analysis is in Sect. 3.3.

3.1 Algorithm Description

The greedy algorithm on uncertain graph is similar to Prim algorithm on exact graph. Specifically, given connective uncertain graph $\mathcal{G} = (V, E, P, W)$, we maintain a tree A, which starts from a random root r and spans an edge at each step until A covers all nodes in V. At each step, a *light edge with largest probability* connecting A and an isolated node in $\mathcal{G}_A = (V, A)$ will be added in A, $\mathcal{G}_A = (V, A)$ is a forest whose node set is same to $\mathcal{G}$, but edge set is A. Initially $\mathcal{G}_A = (V, A)$ is a forest with $|V|$ isolated nodes, with the spanning of A, $\mathcal{G}_A$ adds edges in A increasingly. Intuitively A spans an edge whose one endpoint is in A but the other one is not. *light edge* refers to edge with minimum weight in E, the *light edge with largest probability* is defined as follows.

Definition 3. *Light edge with largest probability(LELP). Add all edges connecting A and isolated node in forest $\mathcal{G}_A = (V, A)$ into queue S, sort S by edge weight in ascending order, for the ith edge in S, say e_i, the probability of adding e_i to A is calculated by Eq. 5, denoted by $\widehat{P}(e_i)$, which is called* join probability*, the edge with largest join probability in S is LELP.*

$$\widehat{P}(e_i) = (\prod_{j=1}^{i-n_i-1} (1 - P(e_j))) \cdot P(e_i) \tag{5}$$

where $1 \leq i \leq |S|$. n_i is the number of edges whose weight is same to the ith edge but position is ahead of it.

The complete algorithm is outlined in Algorithm 2. The input is an uncertain graph $\mathcal{G}$ and the edge set A of MST_{max} is the output. The probability of A is given in Eq. 6, which is the product of the join probability of all edges in A. To gain a better understanding of Algorithm 2, we compute MST_{max} of a simple uncertain graph step by step in the following example.

$$\hat{P}(A) = \prod_{e_i \in A} \hat{P}(e_i) \tag{6}$$

Algorithm 2. Greedy Algorithm

Input: Uncertain graph $\mathcal{G} = (V, E, P, W)$
Output: The edge set A of approximate MST_{max}

```
$MST\_V = \varnothing; A = \varnothing; S = \varnothing; \hat{P}(A) = 1$
Randomly select a root node , say $r$, add it in $MST\_V$
Add all edges connected with $r$ in queue $S$
**while** $|MST\_V| < |V|$ **do**
    Sort $S$ by weight in ascending order
    Calculate the probabality of each edge joinning in $A$ by Eq. 5;
    Get the edge with maximum probabiity by max heap, say $(u, v)$
    Add $(u, v)$ in $A$, suppose $u$ is already in $MST\_V$, $v$ is not,then add $v$ in $MST\_V$
    Update $\widehat{P}(A) = \widehat{P}(A) \cdot \hat{P}(u, v)$
    Delete all edges connecting $v$ in $S$
    Add edges whose one endpoint is $v$ but the other endpoint is not in $MST\_V$ in $S$
return A
```

Example 4. The input uncertain graph $\mathcal{G}'$ is in Fig. 3 with four vertexes and four edges, the weight and existence probability are labeled as binary group on edges.

Initially, we select a node randomly, say a, add a in MST_V and add edges connecting a in S, that is (a, b) and (a, h). sort S in ascending order of weight. We calculate the probability for each edge adding in A using Eq. 4. $\hat{P}(a, b) = P(a, b) = 0.8$, $\hat{P}(a, h) = (1 - P(a, b)) \cdot P(a, h) = 0.12$. We maintain a max heap H to obtain the edge with maximum probability, (a, b) in this case. Add (a, b) in A and the new node b in MST_V.

Next adjust S, delete edges which contains vertex b in S, that is (a, b), then add all edges whose one endpoint is b but the other one is not in MST_V, that is (b, h) and (b, c). Sort S again according to edge weight. Compute the probability of edges in S, $\hat{P}(a, h) = P(a, h) = 0.6$, $\hat{P}(b, c) = P(b, c) = 0.2$, $\hat{P}(b, h) = (1 - P(a, h)) \cdot (1 - P(b, c)) \cdot P(a, b) = 0.224$. The edge with largest probability is (a, h), add it to A and vertex h in MST_V and delete (a, h) and (b, h) in S. Only (b, c) is in S, so we add it in A directly and insert vertex c into MST_V, the MST_{max} edge set $A = (a, b)(a, h)(b, c)$, $\hat{P}(MST_{max}) = \widehat{P}(a, b) \cdot \hat{P}(a, h) \cdot \hat{P}(b, c) = 0.096$.

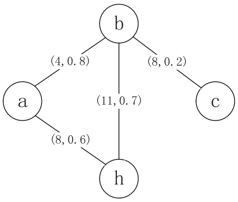


Fig. 3. Uncertain graph G'

3.2 Greedy Selectivity

In this section, we will evaluate the performance of the greedy algorithm we proposed from two aspects, namely, accuracy rate and approximate ratio. To begin with, we analyse the greedy selectivity of our problem.

Suppose we have already known the structure of MST_{max}, that is edges in MST_{max} are given, we can redefine the join probability $\widehat{P}(e_i)$ as Eq. 7. For the ith edge e_i in queue S, we put all edges whose weight is lighter than ei in a set SA_i, $SA_i = \{e_1, e_2 \ldots e_{i-n_i-1}\}$.

$$\begin{aligned} \widehat{P}'(e_i) = P(e_i) \cdot (\prod_{j=1}^{i-n_i-1} (1 - P(e_j)) + \sum_{k=1}^{|SA_i|} \sum_{e_{z1} \ldots e_{zk} \in SA_i} \\ \{\prod_{x=1}^{x=k} P(e_{zx}) \cdot \prod_{\substack{e_m \neq e_{zj} \\ j \in [1,k] \\ e_m \in SA_i}} (1 - P(e_m)) \cdot I_3(e_{z1} \ldots e_{zk})\}) \end{aligned} \tag{7}$$

where $I_3(e_{z1} \ldots e_{zk})$ is a indicator, which indicates whether there exists a path from e_{zx} to e_i in MST_{max} so that $w(e_i)$ is smaller than some edge on that path, $x \in [1, k]$. If no such edge exists, another case in which $I_3(e_{z1} \ldots e_{zk}) = 1$ is that there is no path from e_{zx} to e_i for all $x \in [1, k]$.

$$I_3(e_{z1} \ldots e_{zk}) = \begin{cases} 1 & \exists x \in [1,k], w(e_i) \text{is smaller than} \\ & \text{some edge on the path from } e_{zx} \\ & \text{to } e_i \text{ or } \forall x \in [1,k] \text{ there has no} \\ & \text{path from } e_{zx} \text{ to } e_i \text{ in } MST_{max} \\ 0 & \text{otherwise} \end{cases}$$

We modify line 6 of Algorithm 2 by using Eq. 7 instead of Eq. 5, the other lines remain unchanged. The MST obtained this way is denoted by MST_{new} and the MST obtained by original algorithm is named MST_{old}. The probability of MST_{new} is $P(MST_{new}) = \prod_{e_i \in MST_{new}} \hat{P}'(e_i)$ and we can prove the following theorem is true. We omit the proof due to the page limit.

Theorem 2. *For 2-connected uncertain graph $\mathcal{G} = (V, E, P, W)$, MST_{new} obtained from modified Algorithm 2 is same as MST_{max} in Algorithm 1, that is they have the same edge set and their probability and weight are equal. Formally, $MST_{new} = MST_{max}$, $P(MST_{new}) = P(MST_{max})$ and $W(MST_{new}) = W(MST_{max})$*

Next we will analyse the performance of the approximate algorithm we proposed in Sect. 3.1. Suppose the queue S contains $\{e_1, e_2, e_3 \cdots\}$ in ascending order of their weight currently, then we have the following lemma.

Lemma 1. *If the existence probability of edge obeys uniform distribution in (0,1), then the probability of $\widehat{P}'(e_1) > \widehat{P}'(e_k)$ is at least $\frac{1}{2}$ for $k > 1$.*

Proof. $E[P(e_i)] = \frac{1}{2}$, $P(P(e_i) \geq \frac{1}{2}) = \frac{1}{2}$, $P(P(e_i) < \frac{1}{2}) = \frac{1}{2}$, $\widehat{P}'(e_1) = P(e_1)$ and $\widehat{P}'(e_2) = P(e_2)[(1 - P(e_1)) + P(e_1)I_3(e_1)]$, suppose $P(I_3(e_1) = 0) = p_0$.

The first case is that $P(e_1) \geq \frac{1}{2}$ and $P(e_2) \geq \frac{1}{2}$. If $I_3(e_1) = 0$, $\widehat{P}'(e_2) = P(e_2) \cdot (1 - P(e_1)) < \frac{1}{2}P(e_2) < \frac{1}{2} < P(e_1) = \widehat{P}'(e_1)$. However, if $I_3(e_1) = 1$, $\widehat{P}'(e_2) = P(e_2) > \frac{1}{2}$, then $\widehat{P}'(e_1)$ has $\frac{1}{2}$ probability larger than $\widehat{P}'(e_2)$. Thus the probability in this case is $\frac{1}{2} \cdot \frac{1}{2} \cdot [p_0 + (1 - p_0) \cdot \frac{1}{2}]$. The probability in the other three cases can be computed in a similar way. The total probability of $\widehat{P}'(e_1) > \widehat{P}'(e_2)$ is $\frac{1}{2} + \frac{7p_0}{48}$. Besides, the probability of $\widehat{P}'(e_1) > \widehat{P}'(e_k)$ is obviously larger than $\frac{1}{2}$ for $k > 2$.

Theorem 3. *For 2-connected uncertain graphs* $\mathcal{G} = (V, E, P, W)$*, if the edge probability is independent and identically distributed in* $(0, 1)$ *uniformly, the greedy algorithm can obtain the accurate* MST_{max} *with expected probability at least* $(1 - (\frac{1}{2})^{d/2})^{|V|-1}$.

Proof. In our former analysis, we should select an edge e_i with largest $\widehat{P}'(e_i)$ at each step, so that we can obtain the accurate MST_{max}. However, we apply $\hat{P}(e_i)$ in our approximate algorithm, there are two cases that e_i can still be selected in MST_{max}. The first case is that e_i with largest $\widehat{P}'(e_i)$ also has largest $\hat{P}(e_i)$ among all candidate edges in queue S. The second case is that all indicate function $I_3(e_k) = 0$ for $k \in [1, i-1]$. The expected correct probability for e_i is at least $(\frac{1}{2} + \frac{1}{2} \cdot (1 - (\frac{1}{2})^{(d-1)/2}))$. The detailed proof is omitted due to the page limit.

Theorem 4. *The expected approximate ratio is at least* $(\frac{1}{2})^{d|V|}$.

Proof. We consider the worst case in which $\widehat{P}'(e_i) = P(e_i)$ but $\widehat{P}(e_i) = P(e_i) \cdot \prod_{k=1}^{k=i-1}(1 - P(e_k))$, the approximation ratio is $r = \prod_{k=1}^{k=i-1}(1 - P(e_k))$. Due to $P(e_i)$ is independent random variable, we have $E[\prod P(e_i)] = \prod E[P(e_i)]$, hence $E[r] = E[\prod_{k=1}^{k=i-1}(1 - P(e_k))] = \prod_{k=1}^{k=i-1} E[(1 - P(e_k))] = (\frac{1}{2})^{i-1}$, for $|V| - 1$ edges, the ratio is $r^{|V|-1} > (\frac{1}{2})^{d|V|}$

3.3 Complexity Analysis

In this section, we analyse running time in the worst case. We denote the maximum vertex degree as d, it is obvious to see $1 \leq d \leq |V - 1|$, the length of S in ith iteration is denoted as $|S_i|$, then we have the following relations:

$$\begin{cases} |S_1| \leq d \\ |S_{i+1}| \leq (|S_i - 1|) + (d - 1) \end{cases} \tag{8}$$

The general term formula of arithmetic progression is $|S_i| \leq (d-2) \cdot i + 2 = O(di)$. The total run time is $T(n) = \sum_{i=1}^{|V|}(O(d^2 i) + O(lgdi) + O(1) + O(di) + O(d)) = O(d^2|V|^2)$.

4 Experiments

In this section, we present experimental results studying the effectiveness and efficiency of greedy algorithm.

4.1 Environment and Datasets

Our algorithms were implemented using C++ and the Standard Template Library(STL), and were conducted on a 2.4 GHz Dual Core Intel(R) core(TM) CPU with 2.0 GB RAM running Ubuntu 12.04.

We conduct our experiments on two kinds of synthetic datasets, one of which is generated from real datasets and the other is generated randomly. The first dataset is obtained by assigning a random weight to each edge of real uncertain graphs, the weight is a integer among $[0, 100]$, The real datasets in our experiments are Nature and Flickr, Nature is a protein-protein interaction(PPI) uncertain graph and Flickr is a social network, the scale and connectivity of these two graphs is shown in Table 1, where $N(MST)$ is the number of connected components in graph.

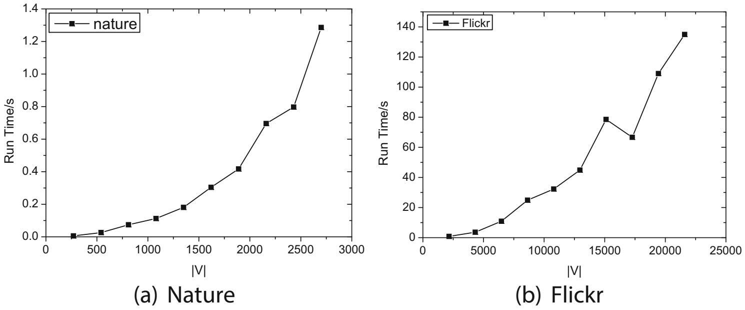
Table 1. Synthetic datasets

Uncertain graph	Graph scale(V,E)	Connectivity	N(MST)
Nature	(2708,7123)	No	63
Flickr	(21594,1008258)	No	1732

The second datasets is generated randomly, to be specific, the edges between vertexes and the edge weight and probability are generated randomly after fixing vertex size, edge weight is a integer among $[0, 100]$ and the probability is among $(0, 0.99]$. We generated three random datasets, the first one is characterized by its average vertex degree, which is 1.23, but the size of vertexes grows from 1k to 10k, we denote this dataset as *man-made1*. The second dataset contains 10 graphs whose vertex sizes are fixed to be 1k, but average vertex degree grows from 3 to 7.5. The third dataset is a set of small uncertain graphs whose vertex size is between 4 to 50 and average degree is 3, we denote it as *man-made3*.

4.2 Analysis of Greedy Algorithm

If the uncertain graph is not connective, we calculate the most reliable forest MSF_{max}. Specifically, when greedy algorithm terminates with a spanning tree A, the vertex size of A is smaller than that of G, we add A in MSF_{max} and randomly select a new root node not in A at the same time, repeat this process until all vertexes in G are covered by MSF_{max}. Next, we analyse the effect of vertex size $|V|$ and average vertex degree $\overline{d}$, as they are two main factor affect the runtime of greedy algorithm.



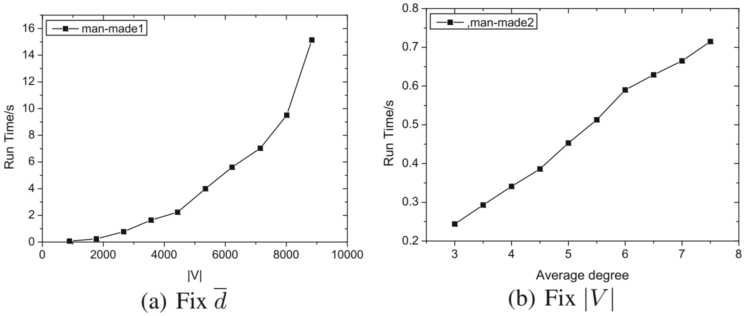
(a) Nature (b) Flickr

Fig. 4. Execution time with different $|V|$

Effect of $|V|$**.** In this experiment, we tested the runtime under different graph scales. We extracted subgraphs from 10 % to 100 % of Nature and Flickr and executed greedy algorithm on subgraphs separately. the results are shown in Fig. 4.

Through the curve in Fig. 4, we can see runtime exhibited parabolically trend increase with the $|V|$, which agrees with our theoretical analysis in Sect. 3.3. However, when we look into Fig. 4(b) carefully we find that the runtime at graph scale (17273,102356) is less than that at (15113,97743) , this is because the average vertex degree $\overline{d}$ of graph (17273,102356) is smaller than that of graph (15113,97743). Therefore we tested the effect of $\overline{d}$ on running time in following experiments.

Effects of $\overline{d}$**.** We ran greedy algorithm on man-made1 dataset, since the average degree on man-made1 is set to be 1.23, we can examine the single effect of $|V|$ on runtime, as shown in Fig. 5(a). The result shows that runtime grows more smoothly when $\overline{d}$ is fixed and no outliers occur. Next we fix $|V|$ and test the effect of $\overline{d}$ on runtime, we apply greedy algorithm on man-made2 dataset whose vertex size is fixed to be 1k and the edge size grows from 3k to7.5k, which means



(a) Fix $\overline{d}$ (b) Fix $|V|$

Fig. 5. Execution time with different $\overline{d}$

$\overline{d}$ is in $[3, 7.5]$, the result of this experiment is shown in Fig. 5(b). We can see that runtime grows linearly as $\overline{d}$ increases when $|V|$ is fixed.

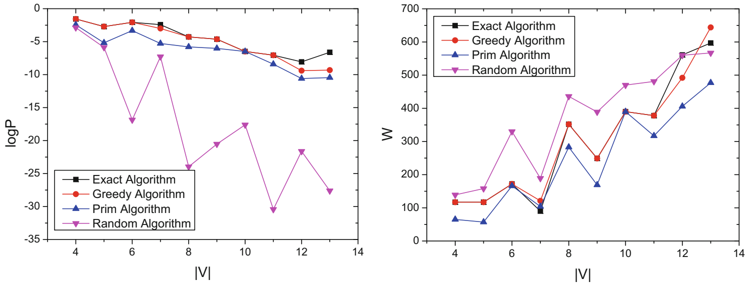
Accuracy. To quantify the accuracy of greedy algorithm, we compare probability and weight of resulting MST with the other three algorithms. We conduct our experiment on man-made3 dataset since the time complexity of exact algorithm is in $O(2^{|E|})$. Furthermore, the probability of MST is very small for graphs whose edge existing probability is relatively small, with the increasing of edges in MST, the probability decreases sharply and it is not convenient for us to record and analyse. Hence we amplify the probability of each edge by computing its log value since log function increases monotonically. Here we have the equation $log \prod_i P_i = \sum_i logP_i$.

We design two contrast experiments to see the effectiveness of our proposed greedy algorithm. The first one is adapted Prim algorithm, which obtains the MST with minimum cost by neglecting the probability on each edge. The other one is a random algorithm, we randomly select one edge and add it in MST.

We compare the probability and weight of MSTs obtained from four algorithms, namely, exact algorithm, greedy algorithm, Prim algorithm and Random algorithm, as shown in Fig. 6(a) and (b). From the figures we can see that the probability and weight of MST obtained by greedy algorithm is same as exact algorithm in most cases. Furthermore, greedy algorithm provides a better approximation to exact solution on probability compared with the other two algorithms.

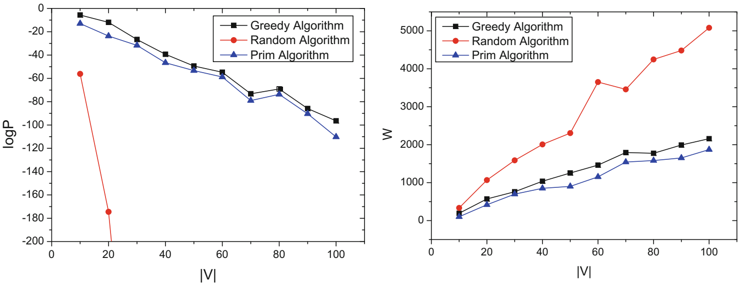
Next we extend our experiment on larger scale graphs without exact algorithm as shown in Fig. 7(a) and (b). We can see greedy algorithm can achieve better probability all the time with a little loss of weight.

Through the experiments conducted above, we come to the following conclusions. First, there are mainly two factors that effect runtime of greedy algorithm, vertex size and vertex degree, furthermore, the grow trend of runtime roughly agrees with our theoretical analysis. Second, our greedy algorithm provides a good approximation to exact algorithm not only on probability but also on weight.



(a) compare logP (b) compare weight

Fig. 6. Compare exact algorithm and greedy algorithm



(a) compare logP (b) compare weight

Fig. 7. Compare greedy algorithm and the other two algorithm

5 Related Work

The minimum spanning tree problem have been studied extensively in the literature under the term stochastic geometry. The main work focus on computing the expected lengths of the MST in stochastic graphs, to the best of our knowledge, there has no previous work on most reliable MST in uncertain graph. To begin with, we survey the work about MST in stochastic graph model, specifically, it mainly includes existential uncertainty model, locational uncertain model and randomly weighted graph model.

Existential Uncertain Model. Given a complete, weighted undirected graph $G = (V, E)$, on n node and m edges, called the master graph, where each node v_i is active(or present) with independent probability p_i. When a node is inactive, all of its incident edges are also absent. We compute the expected minimum spanning tree cost for G, namely, $\sum p(H)MST(H)$, where the sum is over all node-induced subgraphs H of G, $p(H)$ is the probability with which H appears, and $MST(H)$ is the cost of its minimum spanning tree. This problem has been proven to be #P-hard by Kamousi and Suri in [7].

Locational Uncertainty Model. Given a metric space P. The location of each node $v \in V$ in the stochastic graph G is a random point in the metric space and the probability distribution is given as the input. We assume the distributions are discrete and independent of each other. We use p_{vs} to denote the probability that the location of node v is point $s \in P$. The expected length of MST is $E[MST] = \sum_{\mathbf{r} \in R} Pr[\mathbf{r}] \cdot MST(\mathbf{r})$, where $\mathbf{r}$ is a realization of G and can be represented by an n-dimensional vector $(r_1, \ldots, r_n) \in P^n$, where point r_i is the location of node v_i for $1 \leq i \leq n$, $\mathbf{r}$ occurs with probability $Pr[\mathbf{r}] = \prod_{i \in [n]} p_{v_i r_i}$, $MST(\mathbf{r})$ is the length of the minimum spanning tree spanning all points in $\mathbf{r}$. Huang and Li in [8] showed that computing $E[MST]$ in this model is also #P-hard.

Randomly Weighted Graph Model . In this model edge weights are independent nonnegative variables, Frieze and Steele in [9,10] showed that the expected value of the minimum spanning tree on such a graph with identically and independently distributed edges is $\varsigma(3)/D$ where $\varsigma(3) = \sum_{j=1}^{\infty} 1/j^3$ and D is the derivative of the distribution at 0.

Another line is network reliability problem, which computes a measure of network reliability given failure probabilities for the arcs in a stochastic network where each arc can be in either of two states: operative or failed. The state of an arc is a random event that is statistically independent of the state of any other arc. J.Scott has proven that the functional reliability analysis of all-terminal problem is #P-complete in [5].

So far we have quickly reviewed minimum spanning tree problem on stochastic graphs, next we briefly survey problems under the semantic of uncertain. Researchers have studied many kinds of queries on uncertain database, such as Top-k query [12], k-nearest neighbors querey [1], Probabilistic skylines [13]. In addition, lots of work have been done on uncertain graph, including discovering highly reliable subgraphs [14], discovering frequent subgraphs [4] and so on. However, to the best of our knowledge, there is no literature to date on discovering most reliable minimum spanning tree on uncertain graphs. This paper is the first one to investigate this problem.

6 Conclusion

This paper investigates the problem of the most reliable minimum spanning tree on uncertain graph data. The most reliable MST is an optimal choice between stability and cost, which has wide applications in practical. Since accurate algorithms take exponential time to enumerate all possible worlds, an approximate algorithm in polynomial time was proposed to discover an approximate MST and we analysis the accuracy and approximation rate of the approximate algorithm theoretically. The experimental results show that our greedy algorithm has high efficiency and approximation quality.

References

1. Potamias, M., Bonchi, F., Gionis, A., et al.: K-nearest neighbors in uncertain graphs. In: VLDB (2010)
2. Prim, R.C.: Shortest connection networks and some generalizations. Bell Syst. Tech. J. **36**(6), 1389–1401 (1957)
3. Kruskal, J.B.: On the shortest spanning subtree of a graph and the traveling salesman problem. Proc. Am. Math. Soc. **7**(1), 48–50 (1956)
4. Zou, Z., Gao, H., Li, J.: Discovering frequent subgraphs over uncertain graph databases under probabilistic semantics. In: SIGKDD (2010)
5. Provan, J.S., Ball, M.O.: The complexity of counting cuts and of computing the probability that a graph is connected. SIAM J. Comput. **12**(4), 777–788 (1983)
6. Sevon, P., Eronen, L., Hintsanen, P., Kulovesi, K., Toivonen, H.: Link discovery in graphs derived from biological databases. In: Leser, U., Naumann, F., Eckman, B. (eds.) DILS 2006. LNCS, vol. 4075, pp. 35–49. Springer, Heidelberg (2006). doi:10.1007/11799511_5

7. Kamousi, P., Suri, S.: Stochastic minimum spanning trees and related problems. In: ANALCO (2011)
8. Huang, L., Li, J.: Minimum spanning trees, perfect matchings and cycle covers over stochastic points in metric spaces. In: arXiv preprint arXiv (2012)
9. Frieze, A.M.: On the value of a random minimum spanning tree problem. Discret. Appl. Math. **10**(1), 47–56 (1985)
10. Steele, J.M.: On Frieze's (3) limit for lengths of minimal spanning trees. Discret. Appl. Math. **18**(1), 99–103 (1987)
11. Ball, M.O.: Computational complexity of network reliability analysis: an overview. IEEE Trans. Reliab. **35**(3), 230–239 (1986)
12. Soliman, M.A., Ilyas, I.F., Chang, K.C.-C.: Top-k query processing in uncertain databases. In: ICDE (2007)
13. Pei, J., Jiang, B., Lin, X., et al.: Probabilistic skylines on uncertain data. In: VLDB (2007)
14. Jin, R., Liu, L., Aggarwal, C.C.: Discovering highly reliable subgraphs in uncertain graphs. In: SIGKDD (2011)
15. Wu, Y., Fahmy, S., Shroff, N.B.: On the construction of a maximum-lifetime data gathering tree in sensor networks: NP-completeness and approximation algorithm. In: INFOCOM (2008)
16. Manfredi, V., Hancock, R., Kurose, J.: Robust routing in dynamic manets. In: Annual Conference of the International Technology Alliance (2008)

A Block-Based Edge Partitioning for Random Walks Algorithms over Large Social Graphs

Yifan Li[1,2], Camelia Constantin[1], and Cedric du Mouza[2](✉)

[1] LIP6, University Pierre et Marie Curie, Paris, France
{yifan.li,camelia.constantin}@lip6.fr
[2] CEDRIC Lab., CNAM, Paris, France
dumouza@cnam.fr

Abstract. Recent results [5,9,23] prove that *edge partitioning* approaches (also known as *vertex-cut*) outperform *vertex partitioning* (*edge-cut*) approaches for computations on large and skewed graphs like *social networks*. These vertex-cut approaches generally avoid unbalanced computation due to the *power-law degree distribution* problem. However, these methods, like *evenly random assigning* [23] or *greedy assignment strategy* [9], are generic and do not consider any computation pattern for specific graph algorithm. We propose in this paper a vertex-cut partitioning dedicated to random walks algorithms which takes advantage of graph topological properties. It relies on a blocks approach which captures *local communities*. Our split and merge algorithms allow to achieve load balancing of the workers and to maintain it dynamically. Our experiments illustrate the benefit of our partitioning since it significantly reduce the *communication cost* when performing *random walks*-based algorithms compared with existing approaches.

1 Introduction

Random walks-based algorithms, such as personalized PageRank (PPR) [10] and personalized SALSA [4] have proven to be effective in personalized recommender systems due to their scalability. Some recent proposals rely on multiple random walks started from *each vertex* on graph, *e.g.* Fully personalized PageRanks computation using Monte Carlo approximation [3]. We call this intensive computation Fully Multiple Random Walks (FMRWs).

Graph partitioning is a key area of distributed graph processing research, and plays an increasingly important role in both vertex-centric computation, like in *Pregel* model, and query evaluation. Recent results exhibit that *edge partition (vertex-cut)* turned out to be more efficient [5,9] than traditional vertex partitioning (edge-cut) for computation on real-world graphs like social networks. As a consequence, several popular graph computation systems based on this approach have emerged, such as PowerGraph (GraphLab2) [9] and GraphX [23]. However their graph partitioning strategies are generic and do not depend on the algorithms performing the different computations. So they distribute edges evenly over partitions either randomly, *i.e.* a hash function of vertex ids in Giraph [2]

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 275–289, 2016.
DOI: 10.1007/978-3-319-48743-4_22

and GraphX, or using a greedy or dynamic algorithm like in PowerGraph and GPS [19]. Due to the power-law nature of the Web and social network graphs, this edge allocation may lead to an important workload imbalance between the resources. Besides, in contrast with *light-weight* algorithms like PageRank whose messages transmitted between vertices are only rank values, the simulation of *heavy-communication* algorithms, such as *fully (multiple) random walks* in this paper, have a more important communication cost since (*i*) some extra path-related information of walks must also be delivered, and (*ii*) more than one message (walk) start from each vertex at one time. In this case, reducing communication cost is crucial for computation performance guarantee.

We propose in this paper a novel block-based, workload-aware graph(-edge) partitioning strategy which provides a balance edge distribution and reduce the communication costs for random walks-based computations. To the best of our knowledge, this is the first time a partitioning strategy dedicated to fully multiple random walks algorithms is proposed in Pregel model. Finally, the experiments show that our partitioning made significant improvements on both communication cost and time overhead.

Contributions. In summary, our contributions are:

1. a *block-based* partitioning strategy which considers graph algorithms specificities and the topological properties of real-world large graphs along with a seeds selection algorithm for building the blocks;
2. algorithms for merging and splitting blocks to achieve a dynamical load-balancing of the partitions;
3. an experimental comparison of our partitioning approach with several existing random methods over large real social graphs.

After the related work introduced in Sect. 2, Sect. 3 presents our block building strategy while Sect. 4 describes our blocks merge and refinement algorithms. Section 5 presents our experimental results and Sect. 6 concludes and introduces perspectives.

2 Related Work

Pregel [16] has become a popular distributed graph processing framework due to the facilities it offers to the developers for large-graph computations, especially compared with other data-parallel computation systems, *e.g.* Hadoop. Pregel is inspired by *Bulk Synchronous Parallel* [21] computation model where computations on a graph consists of several iterations, also called *super-steps*. During a super-step, each vertex first receives all the messages which were addressed to it by other vertices in the previous super-step. Each vertex performs the actions defined by user-specific function, namely *vertex.compute()* [19] or *vertex.program()* [9], in parallel, using the updated values received in the messages. Then each vertex may decide to halt computing or to pass to other vertices the messages to be used in next super-step. When there is no message transmitted over graph during a super-step (*i.e.* every vertex has decided to halt) the computation stops. Due to *Pregel* success, several optimizations have been recently proposed in literature like the function *Master.compute()* [19] to incorporate global computations or *Mirror Vertices* [14] to reduce communication.

Traditional methods from 2-way cut by local search to multi-level approaches, like Kernighan-Lin [12], PageRank Vectors [1] and METIS-based [11] algorithms, follow a vertex-partitioning (edge-cut) strategy. They propose partitionings which assign (almost) evenly vertices between partitions while minimizing the number of edges cut (edges between two partitions). These algorithms are efficient for small graphs, using in-memory computation. However for real world graphs the large size and the power-law distribution lead to an unbalanced load over edge-cut partitions. More recent partitioning proposals in Pregel-like systems, such as Giraph, GPS, Gelly and Chaos [18] shard the graph using an *edge-cut* strategy which also generates unbalancing for power-law graphs, as introduced in [9].

While there exists a large literature and several implementations for vertex-partitioning, few recent works propose edge-partitioning. The two principal ones are GraphX [23] and PowerGraph [9]. However GraphX only offers random/hash partitioning where edges are evenly allocated over partitions with some constraints of communication between nodes. The underlying graph property, like *local communities in social networks*, is not properly explored. Unlike the hash-like partitioning, PowerGraph uses a heuristic partitioning method, *Greedy Vertex-Cuts*, which has shown significant better performance than random placement in any cases [9]. However, it also ignores the graph topological property and only focus on how to minimize the future communication on previous partitioning situation during edges distribution among partitions. Additionally, unlike our proposal, GraphX and PowerGraph partitionings can not be updated dynamically with graph evolution.

Our approach also takes advantage from the existence of communities. In [8] authors state that, due to the heavy-tailed degree distributions and large clustering coefficients properties in social networks, considering only the direct neighbors of a vertex allows to construct good clusters (communities) with low conductance. In [22] authors improve this method to detect communities over graph, but neither edge partitioning nor workload balancing problem is studied. Moreover, the overlapping communities approach for graph partitioning are not suitable to Pregel-like systems.

3 Block-Based Graph Partitioning

3.1 Principle

Most existing edge partitioning methods, like random [23] or greedy [9] approaches achieve a balanced workload, which means each partition has the same number of edges. Our objective is to go beyond workload balancing and to lower graph processing time by reducing the communication between partitions during graph computation. In *edge-partitioning* approach, a vertex is possibly allocated to multiple partitions and communications between partitions occur when updating the different replicas (mirror vertices) at each Pregel super-step. Consequently, Vertex Replication Factor (VRF) firstly defined in [9] is often used as a communication measurement. So, given an edge partitioning, the communication cost is generally estimated in Pregel, as

$$cost_{Comm} = O(L \times (VRF \times |V|)) \tag{1}$$

where L is the number of supersteps (iterations) during graph computation.

However, in most real graphs, like social networks, there exist many clusters (communities). Our objective is to take advantage of this topological characteristic in our block construction. *Local Access Pattern (LAP)* is described in [24] for first time as one of three kinds of query workload in graphs. We propose to rely on its principles and analysis when proposing our edge-partitioning strategy for random walks-based algorithms considering graph communities to reduce communication costs.

As a consequence we consider that, while VRF is a good estimator of communication cost for some graph algorithms, it is not suitable for the random walks-based algorithms which follow a LAP, since the number of visits of each vertex is different for these algorithms. In other words, communications are conducted unevenly on graph. So our objective is to design a new edge partitioning strategy dedicated to random walks-based algorithms which takes into consideration both the power-law topology of the graph and the LAP characterizing these algorithms.

Our Approach. A block corresponds to a tightly knit cluster in graph, *e.g.* a community in social network. In the *Pregel* approach, we consider the block as a set of edges which are "close" one to another, and these blocks become the component units of each partition in computation, but also the allocation units for workload over machines. Similar to the methodology adopted in vertex partitioning [8,22], we propose to compute a set of K blocks by exploring the graph. An edge is allocated to a block based on its connectivity score from this block. We start a breadth-first search exploration (BFS) from a pre-defined set of K seeds. For each edge encountered we update its connectivity score with respect to all blocks. When the exploration step ended, we allocate the edges to the closest block.

3.2 Connectivity Score of an Edge

In graph computation, how to measure the closeness between a pair of nodes is a fundamental question and it has been studied in many existing works. One interest of these connectivity score measures is to detect cluster in graph (see Sect. 3.3). But based on the observation that for several graph algorithms like random walk, nearest neighbors, breadth-first search, etc., the communications during computations mainly occur between vertices belonging to the same cluster, several approaches extended this cluster detection to perform graph partitioning. For instance [1] proposed a PageRank vector method to find a "good" partition w.r.t. an initial vertex and several pre-set configurations. Besides, there are some proposals like [20] which describes how to obtain these partitions by conducting random walks.

For our edge-partitioning approach, we propose here to estimate the *connectivity score* between an edge and a *query* vertex, e.g. the seed in our paper. We adapt the inverse P-distance [10] used for connectivity score computation between two vertices.

Vertex to Vertex Connectivity Score. Inverse P-distance captures the connectivity: the more numerous and short paths between two vertices, the closer they are in graph topology.

So, the connectivity score $conn_v(i,j)$ from vertex i to vertex j in a directed graph G can be calculated by the paths between them, as follows:

$$conn_v(i,j) = \sum_{p \in P_{ij}} S(p) \tag{2}$$

where the P_{ij} denotes the set of paths from i to j. *S(p)* is the inverse distance score value of path p defines below.

According to the idea of inverse P-distance score, we introduce the concept of "reachability" into connectivity score computation between vertices. The reachability means the probability for a random walk starting from i to arrive at j. So, for path p: $v_0, v_1, ..., v_{(k-1)}$ with length k, *S(p)* can be defined by:

$$S(p) = (1-\alpha)^k \cdot \prod_{i=0}^{k-1} \frac{1}{outDeg(v_i)} \tag{3}$$

where $\alpha \in (0,1)$ is the teleporting probability, *i.e.*, the probability to return to the original vertex, and $outDeg(v_i)$ is the out-degree of vertex v_i.

Vertex to Edge Connectivity Score. Based on the vertex to vertex connectivity score introduced above, we define a vertex to edge connectivity score. We adopt the following definition:

Definition 1 (Edge connectivity score). *The connectivity score* $conn_e(a,b)$ *from a vertex* a *to an edge* $b = (i,j)$ *is:*

$$conn_e(a,b) = \theta(conn_v(a,i), conn_v(a,j))$$

where θ *is an aggregation function.*

In our experiment we choose the average function for θ but other functions like *min* or *max* may also be considered.

3.3 Edges Allocation Algorithm

Based on our edge connectivity score we can now design an edge allocation algorithm. Our algorithm can be decomposed into three steps:

(i) selection of a subset of vertices, namely *seeds*
(ii) connectivity score computation from each edge to all the seeds
(iii) edges allocation to the different blocks

Seeds Selection. We consider for our block-partitioning a seed-expansion strategy: we select a vertex as seed for each block and add each edge to one of the existing blocks. Obviously the result of the partitioning, in term of size-balancing or communication during the computation, is highly dependent on the choice of the seeds. This problem has been studied in literature for instance in [22] to detect communities on graph or in [7] where authors propose and experiment for the pre-computation step of their recommendation algorithm several landmark selection strategies.

Here we adopt the simple but efficient seeds selection procedure, based on *Spread Hubs* method (see [22]), which can be easily deployed on existing graph processing systems. There are two main measurements we used in seeds selection: (1) vertex degree, and (2) connectivity score to other existing seeds. Our seeds selection algorithm is:

1. first we sort the vertices in ascending order, according to their global (in + out) degrees;
2. then we scan the sorted list of vertices, and check if the current one is not *too close* to any existing seed, otherwise we discard it.

The rationale for this algorithm is that a vertex with a large global degree is a vertex with a centrality property and its connected vertices are likely to join its block. Moreover observing a minimum connectivity score between seeds allows a better distribution of the seeds within the graph. Since BFS is efficiently implemented in Pregel, we use it to measure the *distance* between seeds. So we start a BFS from the seed candidate and report the number of hops required to reach the first existing seed. We observe experimentally that we achieve a good partitioning with this algorithm even when the depth of each seed's BFS is set to 1 (so a new seed is not allowed to be the direct neighbor of an existing seed).

Number of Seeds. In our approach, each seed will determine a block which implies to have at least as many seeds as the number of final partitions. However we argue that we can achieve a better partitioning when setting this number to a larger value because:

- the *expansion* of each block can be processed independently, thus can be deployed easily on Pregel-like architecture;
- the combination of small blocks needs *much less* overhead cost than splitting (*i.e.*, refinement) of large blocks when trying to minimize the VRF;
- the more blocks we pre-computed, the higher the level of reusability our partitioning will be.

Connectivity Score Computation. For the second step of our algorithm, we compute first the distance scores of each vertex to all seeds. To perform this connectivity score computation efficiently in our Pregel-like architecture, we proceed to a parallel BFS exploration starting from each seed. Consider a set of seeds $\mathcal{S} = (s_1, s_2, \ldots, s_N)$. We maintain for each vertex ν a connectivity score vector $conn(\nu) = (d_1, d_2, \ldots, d_N)$ where $d_i = conn_v(s_i, \nu)$ is connectivity score to the seed s_i. This vector is updated for each vertex encountered during the BFS exploration.

Since the BFS exploration in large graphs is very costly, we propose to limit the depth of BFS. Indeed we observe in most of the large graphs (like social graphs) a community phenomenon which we capture by selecting the seeds among the vertices with the largest degrees, representing the center of these communities. Intuitively, the distance from the community center is short to other vertices inside the community. Actually, from our experiment results and "Six Degrees of Separation" theory [17], we observe that the *radius of block*, *i.e.*, the connectivity score from seed to potential community members is small and consequently the BFS depth can be set to a small value.

For instance, during the experiment on Livejournal [13] social network, we found the vertex/edge coverages of 200 seeds can reach around 88 % and 96 % by limiting the BFS only to 3 and 4 hops respectively.

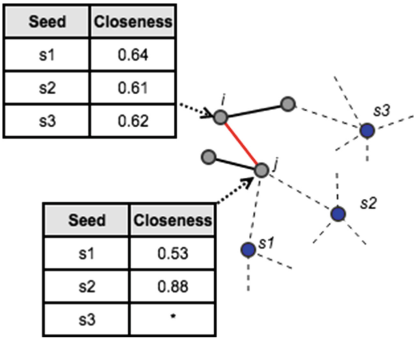
Finally we compute a connectivity score vector for each edge in the graph. Consider an edge $e(\nu, \nu')$ and the connectivity score vectors for its vertices $conn(\nu) = (d_1, d_2, \ldots, d_N)$ and $conn(\nu') = (d'_1, d'_2, \ldots, d'_N)$. Based on Definition 1 we compute the edge connectivity score vector $conn(\varepsilon) = (D_1, D_2, \ldots, D_N)$ as:

$$\forall i \in [1..N],\ D_i = conn_e(s_i, \varepsilon)$$

Edges Allocation. Finally we can allocate the different edges to the blocks according to their edge connectivity score vector. We decide that an edge belongs to the block whose seed is the *closest* to this edge. For edges without any connectivity score value (which means its end vertices have not be reached by any seed during the BFS step), we allocate them in an extra-block.

Example 1. We illustrate the *edge allocation* process with the example in Fig. 1. We assume we have already computed the vertex connectivity score vectors for vertices i and j, considering three seeds s1, s2 and s3. Notice that the '*' value means that the current vertex can not be reached by the seed s_3 in our BFS exploration step. We sum (or make the average) the two vectors to determine the edge connectivity score vector for $e(i, j)$: $conn(i, j) = (0.64 + 0.53, 0.61 + 0.88, 0.62 + 0.0) = (1.17, 1.49, 0.62)$. Here we can clearly point out that the edge e should be allocated to s2 since it has maximum closeness value to this seed.

Observe that some optimizations are possible for storing the vertex connectivity score vectors and for the edge connectivity score vector computation. For instance we can avoid keeping all connectivity score values to every seed, since



Seed	Closeness
s1	0.64
s2	0.61
s3	0.62

Seed	Closeness
s1	0.53
s2	0.88
s3	*

Fig. 1. Example of edge allocation

in this *edge allocation* step, only the maximum value is used to allocate an edge to a block. So we could keep only a *top-k* values for each vertex, with $k \leq |\mathcal{S}|$. Of course the larger k is, the more precise our final result is.

4 Blocks Merge and Refinement Algorithms

Our block partitioning respects the topological properties of the (social) graph, *e.g.* local communities and power-law degree distribution to significantly reduce the communication costs compare to a random allocation strategy.

Given a number of servers P, we must determine how to allocate the different blocks to these servers considering two criteria:

- minimizing the global communication cost;
- balancing the storage and computation workload between servers.

These conditions can be captured by Definition 2.
The first part of the definition allows to control the size of a partition to fit the server capacity and to have an almost balanced edges distribution. The second part of the definition means the partitioning $\mathcal{A}$ is the one which minimizes the *Vertex Replication Factor* (VRF). The VRF measure adopted for instance in [9] means the less partitions the vertex span on average, the less communication across partitions the system initiates for vertices synchronization before running into the next superstep. With respect to Definition 2 we can proceed to the final partitioning based on the different blocks we built.

Definition 2 (Balanced edge partitioning). *Consider a graph G(V,E) where V is the set of vertices and E the set of edges, a set of blocks $\mathcal{B}$ and a number of servers P. The* balanced edge partitioning $\mathcal{A}(\mathcal{B}, P)$ *is defined as:*

$$\mathcal{A}(\mathcal{B}, P) \in 2^{\mathcal{B}},\ such\ that \begin{cases} \forall i \in [1..P], \eta\frac{|E|}{P} \leq |Edge(p_i)| \leq \lambda\frac{|E|}{P} \\ \\ \forall \mathcal{A}' \in 2^{\mathcal{B}}\ satisfying\ above, \\ \frac{1}{|V|}\sum_{v \in V}|alloc(v, \mathcal{A})| \leq \frac{1}{|V|}\sum_{v \in V}|alloc(v, \mathcal{A}')|, \\ \ otherwise,\ relax\ \eta\ and(or)\ \lambda\ to\ find\ the\ \mathcal{A}. \end{cases}$$

where p_i is a partition (server) and $Edge(p_i)$ the edges it contains, $alloc(e, \mathcal{A})$ is the set of partitions to which edge e *is assigned with the partitioning $\mathcal{A}$ (more than one if the vertex is replicated) and $(0 \leq \eta \leq 1 \leq \lambda)$ are small factors to control the storage in each partition.*

Block Split. Since the edges allocation to blocks is only based on a connectivity score criterium some blocks may not fit the maximum size allowed for a partition (second part of Definition 2). Consequently we propose a simple split strategy. Assume that the size of a partition p_i is $(\beta - 1)\lambda\frac{|E|}{P} \leq |Edge(p_i)| < \beta\lambda\frac{|E|}{P}$. We then apply our block building algorithm to the partition p_i with β seeds to split it into β sub-blocks. We potentially iterate the process for any of the sub-blocks which exceeds the partition size.

Blocks Merge. Our block building may also result in producing some blocks whose size is lower than the minimal size (*i.e.* $\eta\frac{|E|}{P}$, see Definition 2). For such a block we re-allocate its edges without considering its seed anymore. Observe that this may lead in turn to some block splits.

Block Allocation. We assume that, possibly after some required splits, the size of all blocks respect the partition size limit. To allocate the blocks to the different partitions, two strategies may be considered: based only on the balancing of the partition sizes, or on minimizing the replication factor between partitions.

Considering this latter approach, we exhibit the following drawbacks: (1) there is an exponential complexity for finding the best blocks allocation considering this criterion, (2) the final size of each partition may highly differ one from another, (3) reducing the global replication factor will not reduce that much the cost of the random-walks algorithms since a path starting in one block and finishing in another is unlikely (according to our blocks building) and finally (4) this partitioning could not evolve dynamically and the partitioning must be re-built when many edges are added or removed.

Consequently we decide to adopt a blocks allocation considering only the size criterion, to achieve a balanced partitioning. We propose a simple but efficient *greedy* algorithm. We allocate the largest block to the partition with the smallest size, and we iterate this strategy until all blocks are allocated. Consequently this allocation is in $O(|\mathcal{B}|)$ where $\mathcal{B}$ represents the set of blocks.

The whole algorithm is presented in Algorithm 1 where *split* refers to a function which proceeds to the block split introduced above, *sortSize* is a function which sorts a set of blocks according to their size, from the largest to the smallest one, and *first* returns the first element from an ordered set.

Managing Graph Dynamicity. Large graphs, especially for social network applications, are often characterized by a high dynamicity. One important aspect of our partitioning algorithm is its ability to manage this dynamicity. Indeed when adding a new edge (for instance when adding a friend on Facebook or an

Algorithm 1. Block allocation algorithm

```
input  : a set $\mathcal{B} = \{b_1, \ldots, b_n\}$ of blocks, a set $\mathcal{P} = \{p_1, \ldots, p_m\}$ of partitions
output: each block is allocated to a $p_j \in \mathcal{P}$
// Initialization to avoid large blocks
$\mathcal{B}' = \emptyset$
foreach $b_i$ in the $\mathcal{B}$ do
    if $b_i.size > \lambda \frac{|E|}{n}$ then
        $\mathcal{B}' = \mathcal{B}' \cup split(b_i)$
    end
    $\mathcal{B}' = \mathcal{B}' \cup b_i$
end
// Sort the set of blocks in descending size order
$\mathcal{B}' = sortSize(\mathcal{B}')$
$b = first(\mathcal{B}')$; while $\mathcal{B}' \neq \emptyset$ do
    $p_i = smallest(\mathcal{P})$;
    $p_i = merge(p_i, b)$; //merge b with the smallest partition
    $\mathcal{B}' = \mathcal{B}' - \{b\}$;
    $b = first(\mathcal{B}')$;
end
Return $\mathcal{P}$ ;
```

url on a Website) we simply have to aggregate the two vertex connectivity score vectors of the two vertices of the edge if both vertices were already present in the graph to compute its edge connectivity score vector. Then we allocate the edge to the block, and consequently to the partition, with the highest connectivity score score. If one of the vertices is new, we have first to perform the BFS exploration from that vertex and compute its vertex connectivity score vector. Potentially this edge allocation may lead to a block split which can be handled with our split algorithm. Oppositely when removing an edge, the size of a block may become too small and we proceed to our block merge algorithm.

5 Experiments

This section presents experiments on our block-based partitioning strategy. We compare it with existing edge partitioning methods: the hash-based approaches [23] and greedy algorithm [9].

5.1 Setting

Computation are performed using GraphX [23] APIs in Spark [25] (version 1.3.1), on a 16 nodes cluster. Each machine has 22 cores with 60 GB RAM running Linux OS. For our experiments we set teleporting probability α to a classical value 0.15. The depth of the BFS exploration (*i.e.*, the maximum length considered for paths from seed to other vertices).

Data Sets . We validate our approach on two datasets: LiveJournal [6] with 4.8M vertices and 68.9M edges, and Pokec [15] with 1.6M vertices and 30.6M edges. These datasets can be downloaded from *SNAP* [1].

Competitors . *Hash Partitioning.* There are four wide used random(hash)-like partitioning methods[2], introduced in GraphX:

- RandomVertexCut: allocates edges to partitions by hashing the source and destination vertex IDs.
- CanonicalRandomVertexCut: allocates edges to partitions by hashing the source and destination vertex IDs in a canonical direction.
- EdgePartition1D: allocates edges to partitions using only the source vertex ID, co-locating edges with the same source.
- EdgePartition2D: allocates edges to partitions using a 2D partitioning of the sparse edge adjacency matrix.

Greedy Vertex-Cuts. PowerGraph proposes a greedy heuristic for edge placement process which relies on the previous allocation of vertices to determine the partition next edge should be assigned.

5.2 Communication

Our approach aims at reducing the runtime graph processing thanks to a significant reduction of the communication costs.

Vertex Replica Factor (VRF). VRF is the traditional way to compare two partitionings regarding the communication costs, independently of the algorithm executed. We compare the VRF of our *Block-based* partitioning with the one of the competitors for different numbers of partitions. Results are depicted on Fig. 2. We observe that, as observed in [9], partitioning strategies based on topology outperform as expected hash-based methods: VRF decreased by 30–60 % (resp. 60–80 %) for Powergraph (resp. our block strategy) compare to the strategies used in GraphX. This experiment also illustrates the benefit of our global approach for edge allocation compare to a greedy approach with on average a 40 %-lower VRF.

Number of Messages. VRF is a general criterion to compare two partitioning strategy independently from the algorithms, but we expect our partitioning to exhibit even better results for random walks-based algorithms. Consequently to estimate the benefit of our approach we simulate fully multiple random walks (FMRW) and we measure the number of messages exchanged between partitions. From each vertex we perform 2 random walks of length 4 and we report

[1] https://snap.stanford.edu/data/index.html.

[2] See details and implementations at http://spark.apache.org/docs/latest/api/scala/index.html.

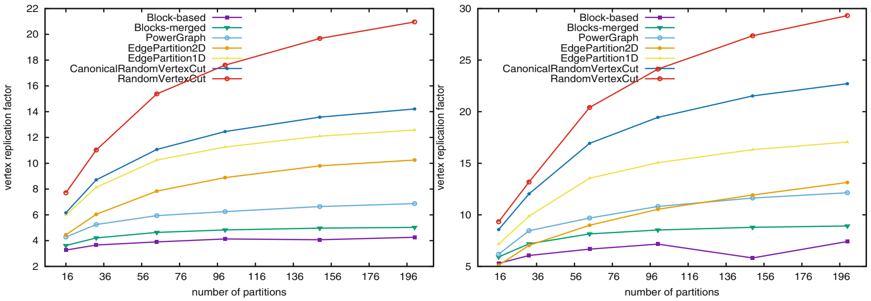


Fig. 2. VRF w.r.t. edge partitioning methods on LiveJournal (left) and Pokec (right)

experimental results in Table 1. We observe that our method reduces significantly the number of messages exchanged between partitions. For instance with 100 partitions, 61.8 million messages are necessary for processing the FMRW with our method while 381.9 million are transmitted with Random-Vertex-Cut method, so a drop of 84 %. This result was expected since the VRF is 3–4 times lower with our method than with Random-Vertex-Cut. But we notice that if the reduction of the number of messages and of the VRF were proportional, the system should exchange 89.4 million message. This 30 % gain in the number of messages transmitted validates our intuition that random walks intend to stay in the local cluster (community). So low-replicated vertices (close to the seed in block) are accessed more times, and oppositely few random walks reach the farthest, high-replicated, vertices. Similar results are obtained from experiments on Pokec.

Table 1. Messages transmitted in FMRW (LiveJournal)

#Partitions	Random-vertex-cut [23]		Block-based partitioning			
	VRF	Real mess	VRF	Real mess	Expected mess	Ratio
64	15.38	303.5 m	3.90	55.3 m	76.9 m	0.72
100	17.61	381.9 m	4.13	61.8 m	89.4 m	0.69
150	19.68	464.8 m	4.07	70.6 m	96.1 m	0.73
200	21.12	525.6 m	4.26	76.0 m	106.0 m	0.72

5.3 Runtimes

We propose to evaluate how the runtime of different graph processing algorithms benefits our partitioning, compared to other methods. First, we launch FMRW, a heavy-communication algorithm, on LiveJournal and Pokec datasets respectively, with 3 random walks of length 4 started from each vertex. From the results

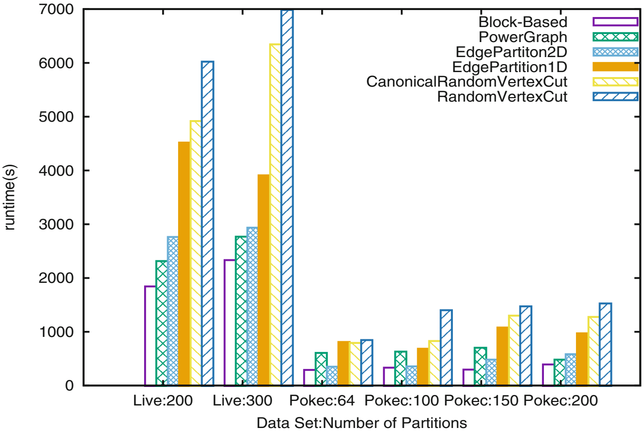


Fig. 3. Runtimes for FMRW with different partitionings for LiverJournal and Pokec

in Fig. 3, we see that our partitioning can save up between from 20 to 65 % of runtime, compared with other partitionings, for both datasets.

We also test our method with traditional PageRank algorithm. We consider the static (fixed number of iterations) and dynamical (with convergence and a threshold value) approaches. We consider there are 200 partitions and we proceed to resp. 30, 50 and 100 iterations for static PageRank and to dynamical PageRank with resp. 0.005 and 0.001 convergence factor. Figure 4 depicts results and confirms that our partitioning method outperforms other ones. While we observe a small 5–20 % gain for the static implementation of PageRank, we reach a 20–55 % gain for the dynamical implementation.

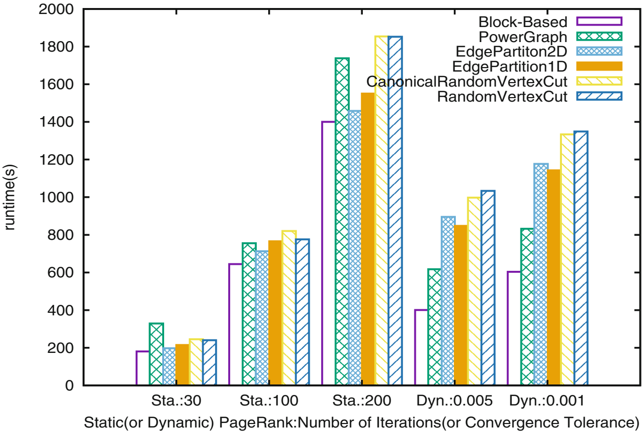


Fig. 4. Runtimes for static and dynamic PageRank for LiverJournal

6 Conclusion and Future Work

We present in this article a vertex-cut partitioning for random-walks-based algorithms relying on the topology to build blocks which respect local communities. We propose *split* and *merge* algorithms to get and to maintain the final partitioning. We experimentally demonstrate that our proposal outperforms existing solutions.

As future work we plan to investigate different seeds selection algorithms. While this problem has been studied in different contexts (see [7,22]) we believe that the nature of the graph algorithms, here random walks-based algorithms, must be considered when selecting the seeds. We also intend to study the 5–10 % of vertices which are not reached by the BFS exploration issued at seeds. They are located on the periphery of social graph and are poorly connected. While we currently place them to an extra-block, we will design a strategy to allocate them to existing blocks.

References

1. Andersen, R., Chung, F., Lang, K.: Local graph partitioning using PageRank vectors. In: FOCS, pp. 475–486 (2006)
2. Apache. Giraph. http://giraph.apache.org
3. Bahmani, B., Chakrabarti, K., Xin, D.: Fast personalized PageRank on MapReduce. In: SIGMOD, pp. 973–984 (2011)
4. Bahmani, B., Chowdhury, A., Goel, A.: Fast incremental and personalized PageRank. Proc. VLDB Endow. **4**(3), 173–184 (2010)
5. Bourse, F., Lelarge, M., Vojnovic, M.: Balanced graph edge partition. In: SIGKDD, pp. 1456–1465 (2014)
6. Chierichetti, F., Kumar, R., Lattanzi, S., Mitzenmacher, M., Panconesi, A., Raghavan, P.: On compressing social networks. In: SIGKDD, pp. 219–228 (2009)
7. Dahimene, R., Constantin, C., du Mouza, C.: RecLand: a recommender system for social networks. In: CIKM, pp. 2063–2065 (2014)
8. Gleich, D.F., Seshadhri, C.: Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In: SIGKDD, pp. 597–605 (2012)
9. Gonzalez, J.E., Low, Y., Gu, H., Bickson, D., Guestrin, C.: PowerGraph: distributed graph-parallel computation on natural graphs. In: OSDI, pp. 17–30 (2012)
10. Jeh, G., Widom, J.: Scaling personalized web search. In: WWW, pp. 271–279 (2003)
11. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. **20**(1), 359–392 (1998)
12. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. Bell Syst. Techn. J. **49**(2), 291–307 (1970)
13. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters. Internet Math. **6**, 29–123 (2008)
14. Low, Y., Bickson, D., Gonzalez, J., Guestrin, C., Kyrola, A., Hellerstein, J.M.: Distributed GraphLab: a framework for machine learning and data mining in the cloud. VLDB Endow. **5**(8), 716–727 (2012)

15. Lubos Takac, M.Z.: Data analysis in public social networks. In: Present Day Trends of Innovations, pp. 1–6 (2012)
16. Malewicz, G., Austern, M.H., Bik, A.J.C., Dehnert, J.C., Horn, I., Leiser, N., Czajkowski, G.: Pregel: a system for large-scale graph processing. In: SIGMOD, pp. 135–146 (2010)
17. Newman, M., Barabasi, A.-L., Watts, D.J., Structure, T.: Dynamics of Networks: (Princeton Studies in Complexity). Princeton University Press, Princeton (2006)
18. Roy, A., Bindschaedler, L., Malicevic, J., Zwaenepoel, W.: Chaos: scale-out graph processing from secondary storage. In: SOSP, pp. 410–424 (2015)
19. Salihoglu, S., Widom, J.: GPS: a graph processing system. In: SSDBM, pp. 22:1–22:12 (2013)
20. Sarkar, P., Moore, A.W.: Fast nearest-neighbor search in disk-resident graphs. In: SIGKDD, pp. 513–522 (2010)
21. Valiant, L.G.: A bridging model for multi-core computing. J. Comput. Syst. Sci. **77**(1), 154–166 (2011)
22. Whang, J.J., Gleich, D.F., Dhillon, I.S.: Overlapping community detection using seed set expansion. In: CIKM, pp. 2099–2108 (2013)
23. Xin, R.S., Gonzalez, J.E., Franklin, M.J., Stoica, I.: GraphX: a resilient distributed graph system on spark. In: GRADES, pp. 1–6 (2013)
24. Yang, S., Yan, X., Zong, B., Khan, A.: Towards effective partition management for large graphs. In: SIGMOD, pp. 517–528 (2012)
25. Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M.J., Shenker, S., Stoica, I.: Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. In: NSDI, p. 2 (2012)

Differentially Private Network Data Release via Stochastic Kronecker Graph

Dai Li[1], Wei Zhang[1,2](✉), and Yunfang Chen[1]

[1] College of Computer, Nanjing University of Posts and Telecommunications,
Nanjing, China
zhangw@njupt.edu.cn

[2] Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks,
Nanjing 210003, Jiangsu, China

Abstract. Excessive sensitivity problem due to complication of data has been a non-negligible challenge to data privacy protection under differential privacy recently. We design a private data release framework called DPDR-SKG (Differentially Private Data Release via Stochastic Kronecker Graph), which focuses on releasing social network data under differential privacy and uses a two-phase privacy budget allocation. Firstly, we cluster the similar communities of network according to Stochastic Kronecker graph parameter. Secondly, we implement optimized privacy budget allocation in terms of cluster distribution. Experimental results show that the DPDR-SKG outperforms in preserving the privacy of network structure and effectively retaining the data utility.

Keywords: Social network graph · Differential privacy · Privacy budget · Network model · Community structure

1 Introduction

Social network data contain lots of valuable and sensitive information. Since Samarti and Sweeney obtained the governor's health information in 1998 by connecting Massachusetts health information table with the voter registration form, personal privacy protection has become the social focal topic [2]. For traditional structural data, the privacy protection models often appear very weak in the face of attacks based on background knowledge. Differential privacy was firstly proposed by Dwork [1] in 2006. It is a new kind of definition for statistical database privacy disclosure issues that whether a single record exists in the data set or not, has almost negligible impact on the calculation results. So, the privacy risks are controlled within the tiny and acceptable scope. The algorithm utilizes parameter ε to denote privacy budget or privacy protection level. The mechanism of differential privacy protection is adding an appropriate amount of interference noise to the query function return value. Existing privacy budget

W. Zhang—Project supported by the National Natural Science Foundation of China under grants 61272422, 61202353.

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 290–297, 2016.
DOI: 10.1007/978-3-319-48743-4_23

allocation methods often can't reflect social network properties and structure. DPDR-SKG is inspired by community features of the social network, then optimizes privacy budget allocation methods, eventually releases network data which satisfies differential privacy.

2 Related Work

Faced with various attacks and analysis methods, some privacy protection methods were proposed, such as K-anonymity [2], but the defects of these protection models gradually revealed because They overly depend on the assumptions of the background knowledge and cannot provide quantitative analysis. In order to solve above problems, Dwork in 2006, brought up the strict concept of differential privacy, privacy risks are controlled within tiny and quantifiable scope, so differential privacy has become a prime choice in data privacy research. Since rigorous concept does not apply to complex data in reality, Dwork et al. [3] presented the extensive optimization concept of global sensitivity: smooth sensitivity for practical application. Zhu [4] defined a new form of sensitivity, the correlated sensitivity.

Social network data release under differential privacy is divided into interactive and non-interactive release. The current data research focuses on the non-interactive release. Meanwhile Karwa and Slavković [5] also studied the network degree sequence differential privacy. So as to better protect the structure of network properties, Sala et al. [6] developed a differentially private graph model called Pygmalion, which can extract graphs structure into degree correlation statistics DK series. DK series maintains as much structural similarity to G as possible, while supporting abundant sub-graph counting queries. Xiao et al. [7] claimed that social network data can be converted to Hierarchical Random Graph (HRG) model, which mixes network structure and statistical inference. Lus work [8] suggests that social network graph can be converted to several parameters of Exponential Random Graph Model (ERGM). Generally protecting network parameters from identification can help protect the privacy of the entire network. Mir and Wright [9] introduced maximum likelihood estimation algorithm to SKG graph model. Our paper also uses SKG model to analyse community structure instead of whole network.

3 Preliminaries

3.1 Definition of Differential Privacy

Definition 1 (Differential Privacy): A randomized algorithm M with domain $N^{|\chi|}$, is (ε, δ)-differentially private for all $S \subseteq Range(M)$ and for all $x, y \in N^{|\chi|}$, the randomized mechanism satisfies:

$$\Pr[M(x) \in S] \leq \exp(\varepsilon) \Pr[M(y) \in S] + \delta \tag{1}$$

The sensitivity of a function Δf is defined as the maximum difference in function output when one single domain is modified.

Definition 2 (Neighboring Dataset): Given two dataset D and D', they share the same structure and property, $|D\Delta D'|$ indicates the number of records they differ, once $|D\Delta D'|=1$, we call D and D' neighboring datasets.

Definition 3 (Global Sensitivity): For $f : D \to R^d$, the global sensitivity of f is defined as

$$GS_f = \max_{D,D'} \|f(D) - f(D')\|_1 \tag{2}$$

3.2 SKG Model

Parametric network model assumes that the observed data is generated by a series of probability estimation, $\mathrm{P} = \{f(x,\theta) : \theta \in \Theta\}$, in which θ represents the unknown parameters or vectors from the value space Θ. In this way, the structure and features of network need to be protected by defending the parameters. In the paper, we select SKG (Stochastic Kronecker Graph) model. The graph model effectively captures some key patterns of real-world graphs and can deal with very large and complex networks due to the fast and scalable MLE algorithm.

Definition 4 (Stochastic Kronecker Graph): Given an $N \times N$ matrix: Θ, $\theta_{i,j} \in \Theta$ represents the probability that edge (i,j) exists in the graph, $\theta_{i,j} \in (0,1)$. Then the k-th Kronecker power $P = \Theta^{[k]}$, is a stochastic matrix where each value $p_{u,v} \in P$ encodes the probability of edge (u,v) appearing. A stochastic graph is generated from the stochastic Kronecker matrix. The calculation of $p_{u,v}$ is as follows:

$$p_{uv} = \prod_{i=0}^{k-1} \Theta\left[\left\lfloor \frac{u-1}{N^i} \right\rfloor (\mathrm{mod} N) + 1, \left\lfloor \frac{v-1}{N^i} \right\rfloor (\mathrm{mod} N) + 1\right] \tag{3}$$

Gleich and Owen [10] propose the so-called moment-based estimation of the model parameter, we refer model parameters of the original graph from the observed statistics. Four statistics for matching are considered: number of edges (E), number of triangles (Δ), number of 2-stars (H) and number of 3-stars (T). They consider graphs with a $2 * 2$ initiator matrix of the form $\hat{\Theta} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$

Definition 5 (SKG Function): Given an undirected graph G, according to the SKG model, we get the definition of SKG function:

$$SKG(G) : G \to \hat{\Theta} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \tag{4}$$

4 Differentially Private Data Release via SKG

Since one social network has been divided into some communities, we convert each community to a single SKG parameter. And then we cluster the communities with the similar SKG parameters into the same group. Afterward, we allocate the privacy budget in the groups according to the number of communities in each group.

4.1 Clustering Communities

We replace a SKG parameter vector of the whole network with a set of SKG parameters vector of the communities, whose advantages are listed below (1) Network structure is better preserved because community structure is simpler and more distinctive than whole network structure; (2) Community treatment ignores the influence of the bridge edges between the communities, and the independence of the sensitivity calculation will be proved in the following section; (3) DPDR-SKG decomposed the parameters model privacy protection into division, each local divisions satisfies differential privacy, so does the whole network.

It is obvious that similar parameters infer the similar network structures, meanwhile SKG parameters are determined by initial matrix. As a consequence, similar initial matrix denotes analogical network structure. The proposed clustering process puts the similar communities indicated by the similar $\hat{\Theta}$, in one group, and the similarity is decided by the Euclidean distance of $\hat{\Theta}$.

Definition 6 ($\hat{\Theta}$ Euclidean Distance): Given a SKG matrix of sizes $N \times N$, $\Theta = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$, Θ_x, Θ_y have the same size, the Euclidean Distance is defined below:

$$d_{x,y} = \sqrt{\sum_{j=1}^{n} \sum_{i=1}^{n} (a_{i,j} - a'_{i,j})^2} \tag{5}$$

We adopt a threshold clustering approach that calculates the Euclidean distance between all parameters and form a distance matrix. After the approach, we can get

$$\{\hat{\Theta}_1, \hat{\Theta}_2, \ldots, \hat{\Theta}_n\} \rightarrow \{\Gamma_1, \Gamma_2 \ldots \Gamma_k\}, k <= n \tag{6}$$

so n communities are clustered into k groups and the distance of any pair communities in one group is less than the threshold.

4.2 Allocating Privacy Budget

The core of the two phase budget allocation is the more communities the group has, the less privacy budget the group needs. In the first phase, privacy budget allocation is mainly based on the number of communities in each group. According to the parallel composition, we assume the whole privacy budget is ε. In the worst situation, we assign the budget to the group who has the least number of communities to make sure that the whole network data are protected under differential privacy. Afterwards, we allocate privacy budget to other groups due to the numbers of communities inside. According to our method, we have the following equations, n_{min} denotes the minimum number of communities in single group:

$$\varepsilon_{\Gamma_i} = \varepsilon . \frac{n_{min}}{n_{\Gamma_i}} \tag{7}$$

Table 1 shows the example of privacy budget allocation. The whole network is divided into 4 groups, consisting of 1, 4, 7, 8 communities. Group A has only 1 community, least number among other groups. According to our method, the budget of Group B is computed, $\varepsilon_B = \varepsilon . \frac{n_{min}}{n_{\Gamma_i}} = \frac{\varepsilon}{4}$, group C and group D get privacy budget $\frac{\varepsilon}{7}$ and $\frac{\varepsilon}{8}$.

In the second phase, the privacy budgets are divided based on the community unit. The sensitivity calculation of each community is independent ,the proof is introduced in Sect. 4. We just need to inject some noise to the key statistics to preserve parameter privacy.we give considerations to the structure and numbers of communities inside the group. Each community receives the same privacy budget on account of the similar parameters.

Table 1. Example of privacy budget allocation

Groups	Group A	Group B	Group C	Group D
Community numbers	1	4	7	8
Group budget	ε	0.25ε	0.143ε	0.125ε
Community budget	ε	0.25ε	0.143ε	0.125ε

4.3 Differentially Private Algorithm

We compute a differentially private estimator based on the above theories. Whole privacy budget is divided into 2 parts for private approximation to parameters. One is for injecting noise to E, H, and T by computing approximation to the degree sequence vector of G and the other is for adding noise to the number of triangles in G respectively. Finally, we generate the synthetic graphs using the output estimator and release the data to the public for research purposes.

Lemma 1. *DPDR-SKG meets the definition of edge differential privacy*

Proof. Two situations are discussed below: (1) Edge privacy within communities; (2) Edge privacy between communities. According to [10], the SKG parameters are decided by 2 statistics, which are degree sequence and the numbers of triangle. We need to discuss edge privacy analysis on degree sequence and triangle numbers. The method to inject noise to degree sequence was proposed in [11]. Given an undirected graph community G_C, let d be the degrees sequence of G_C, Hay proposed a method of computing a differentially private approximation d^* of the sorted degree vector d_S by adding a vector of appropriate Laplace noise. Therefore, we compute $(\epsilon, 0)$ -differentially private approximations of E', H', T'. Injecting noise to the numbers of triangle adopts the methods mentioned in [3]. The smooth sensitivity can be used to compute a (ε, δ)-differentially private approximation of Δ. In conclusion, we prove that the computation of (E', H', T', Δ') meets $(2\varepsilon, \delta)$ differentially private within the community.

Assuming that d_{Ci} represents the degree sequence of i-th community, d_{Cj} denotes the j-th community. The proof proceeds by the following the hypothesis: The sensitivity of d_{Ci} is influenced by the changes of d_{Cj}. The augment or removal of single edge only affects the degrees of 2 corresponding nodes. Assuming node m, which has x edges, is in community C_i, node n, which has y edges, is in community C_j, if we add one edge between node m and node n, the degree of m and n becomes $x+1$, $y+1$, the sensitivity of degree sequence of community C_i changes to 1, so does community C_j. The above results are opposite to the hypothesis. Therefore we prove that sensitivity calculation of degree sequence is independent. Triangle consists of three vertices and corresponding 3 edges between vertices. If we remove or add an edge across 2 communities, it has no effect for internal triangle numbers of two communities respectively.

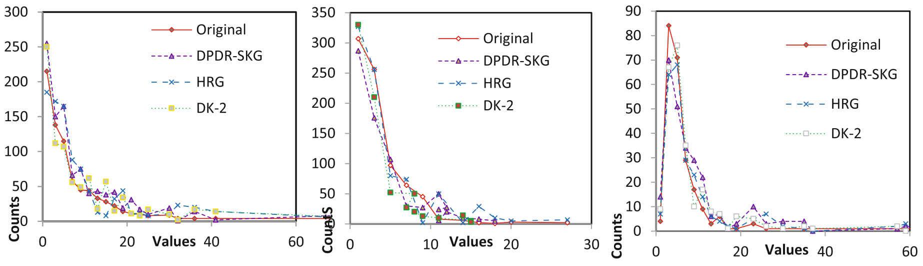
So we conclude that sensitivity calculation of triangle number is also independent. Through the above analysis, our two-phase method definitely meets the definition of differential privacy. □

5 Experiment Evaluation

In this session, we mainly introduce the performance of proposed methods applied in real datasets. We select 3 real-world data sets, which are WBLOGS, NetScience [12] and BIO [13]. WBLOGS represents the social network graph data. NetScience and Bio-celegans contain a co-authorship network. All data are pre-processed into undirected graph whose edges are un-weighted without self-loops.

In the experiment, we compare DPDR-SKG with other data release approaches, such as HRG and DK-2 models, as well as the original graph data. Due to the randomness, we examine the variances of its performance by running the algorithm ten times on each network and compute various expected statistics. In the following figures, Original refers to the original graph, HRG and DK-2 represents the data release method proposed by [6,7] respectively. We summarize 3 statistics briefly, Degree distribution, Clustering coefficient and Eccentricity distribution.

Figure 1 shows the degree distributions of the released data under privacy budget $\varepsilon = 1$. It can be seen that, DPDR-SKG protects the properties of both



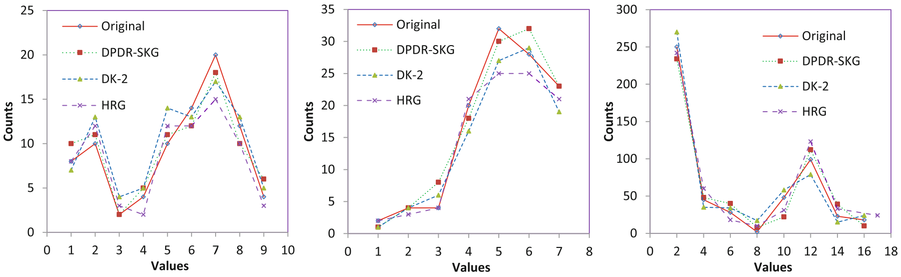
(a) WBLOGS (b) NetScience (c) BIO

Fig. 1. Degree distribution, $\varepsilon = 1$

high-degree nodes and low-degree nodes better, comparing with other cases. From the above figures, we observe that released data by 3 methods all perform well and match the original graph.

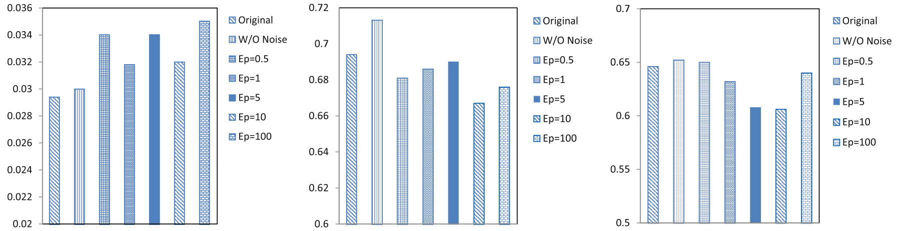
Figure 2 compares the eccentricity centrality for each graph data. We observe that our method efficiently captures the important nodes locating in the center of the network. By observing our method, we can also see the decrease of long path length. We believe that is due to the influence of edges changed between the communities, but this does not have a big influence on the structure of network.

Figure 3 indicates the average clustering coefficient of released data for each network. We only focus on our method and want to find out how our method responds to different privacy budgets. To be fair, we also plot the graph data without injecting noise, and change $\varepsilon = 0.5\ 1\ 5\ 10\ 100$. It can be observed that the average clustering coefficient of released data matches closely with original graph with small privacy budget. When we increase the privacy budget, the accuracy varies greatly and the statistics arent stable and robust, which destroys the utilization of released data. For privacy concerns, it is better to limit privacy budget to small scale, our method performs well with robustness.



(a) WBLOGS (b) NetScience (c) BIO

Fig. 2. Eccentricity distribution



(a) WBLOGS (b) NetScience (c) BIO

Fig. 3. Average clustering coefficient

6 Conclusions

We propose a graph mechanism to release graph data while protecting individual privacy. The results of experiments show that our method not only effectively protect the original structure of the network graph, but also guarantee the utilization of released data. There are several research directions for future work. For example, we intend to find more suitable graph models, such as ERGM. Furthermore, the privacy allocation strategy can still be optimized to achieve higher accuracy.

References

1. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006)
2. Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **10**(05), 571–588 (2002)
3. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006)
4. Zhu, T., Xiong, P., Li, G., Zhou, W.: Correlated differential privacy: hiding information in non-IID data set. IEEE Trans. Inf. Forensics Secur. **10**(2), 229–242 (2015)
5. Karwa, V., Slavković, A.B.: Differentially private graphical degree sequences and synthetic graphs. In: Domingo-Ferrer, J., Tinnirello, I. (eds.) PSD 2012. LNCS, vol. 7556, pp. 273–285. Springer, Heidelberg (2012)
6. Sala, A., Zhao, X., Wilson, C., Zheng, H., Zhao, B.Y.: Sharing graphs using differentially private graph models. In: Proceedings of 2011 ACM SIGCOMM Conference on Internet Measurement Conference, pp. 81–98. ACM (2011)
7. Xiao, Q., Chen, R., Tan, K.L.: Differentially private network data release via structural inference. In: Proceedings of 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 911–920. ACM (2014)
8. Lu, W., Miklau, G.: Exponential random graph estimation under differential privacy. In: Proceedings of 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 921–930. ACM (2014)
9. Mir, D.J., Wright, R.N.: A differentially private graph estimator. In: IEEE International Conference on Data Mining Workshops, ICDMW 2009, pp. 122–129. IEEE (2009)
10. Gleich, D.F., Owen, A.B.: Moment-based estimation of stochastic Kronecker graph parameters. Internet Math. **8**(3), 232–256 (2012)
11. Hay, M., Li, C., Miklau, G., Jensen, D.: Accurate estimation of the degree distribution of private networks. In: 9th IEEE International Conference on Data Mining, ICDM 2009, pp. 169–178. IEEE (2009)
12. Rossi, N.A.R.: Network repository (2012–2016). http://networkrepository.com/index.php
13. DuBois, C.L.: UCI network data repository (2008)

An Executable Specification for SPARQL

Mihaela Bornea[1], Julian Dolby[1(✉)], Achille Fokoue[1],
Anastasios Kementsietsidis[2], Kavitha Srinivas[1], and Mandana Vaziri[1]

[1] IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA
{mbornea,dolby,achille,ksrinivs,mvaziri}@us.ibm.com
[2] Google Research, Mountain View, CA, USA
anastasios@alumni.utoronto.ca

Abstract. Linked Data on the web consists of over 1000 datasets from a variety of domains. They are queried with the SPARQL query language. There exist many implementations of SPARQL, and this rich ecosystem has demanded a precise specification and compliance tests. However, the SPARQL specification has grown in complexity, and it is increasingly difficult for developers to validate their implementations. In this paper, we present a declarative specification for SPARQL, based on relational logic. It describes SPARQL with just a few operators, and is executable: queries written in it can be directly executed against real datasets.

Linked Data on the web consists of over 1000 datasets from a variety of domains: curated medical literature [14], geographical data [6], financial data [13], crowd-sourced general knowledge [4], government data [3] and many more. Across many domains, programming against datasets increasingly involves querying with the SPARQL query language [10,12].

However, the SPARQL specification is *operational*, i.e., given by translating the language to an algebra, and is non-trivial for complex queries. We present an alternative, declarative specification for SPARQL based on *relational logic*, where relations are first-class entities and can be composed via relational and logical operators to form *constraints*. A *solution* to a relational constraint is an assignment of data to the relations that satisfies the constraint. The graph-based nature of RDF is a natural fit for relational logic: subject-predicate-object expressions are represented as relations and the meaning of queries is given in terms of relational operators. Our specification is *declarative*, i.e., the structure of SPARQL queries is mapped directly to relational constraints whose solutions represent answers to the query.

We have validated our specification by applying the compliance tests provided by SPARQL. This was done by writing our relational-loic semantics in Kodkod and running it on compliance tests.

1 Semantics

In this section, we give meaning for the SPARQL core. We cover the graph patterns as defined in the specification (see [10], Sects. 5–10 and 19.8, grammar rules 53–68): basic graph pattern (BGP), union, optional, filter, minus, bind and values.

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 298–305, 2016.
DOI: 10.1007/978-3-319-48743-4_24

The meaning of a query Q for a dataset D is the set of bindings of the variables of Q such that the denoted subgraph appears in D. $\mathfrak{T}(Q, D)$ denotes the formula in the free variables of Q that satisfies Q in D; $\mathfrak{T}$ is defined in Fig. 3. The semantics of graph patterns makes use of expressions in some cases, which is defined in Fig. 4.

1.1 Graph Patterns

We introduce and motivate $\mathfrak{T}$ with a series of examples that cover the core.

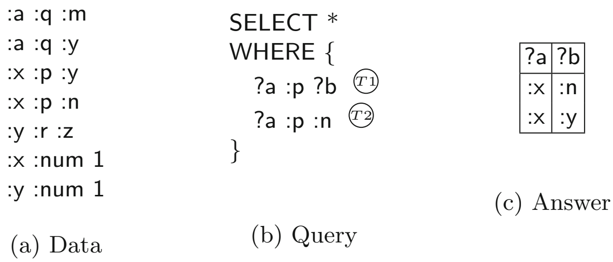
Basic Graph Patterns. An example dataset is shown in Fig. 1(a), along with a simple SPARQL query in Fig. 1(b). This is a basic pattern specifying two variables ?a and ?b, and matches pairs of triples where a binding for ?a has :p edges to ?b and to :n. Given the dataset on the left, the query in the middle evaluates to the answer on the right, Fig. 1(c), since `:x` has a `:p` relation to `:n`, which can satisfy the second triple pattern (T2). Both `:n` and `:y` can satisfy the first triple pattern (T1). Our basic approach is to express the query as a logical formula. For a triple $\langle a, b, c\rangle$, we simply require the triple be in the dataset: $\mathfrak{T}(\langle a, b, c\rangle) \equiv \langle a, b, c\rangle \in G$. For the join between the two triples, we use logical and: $\mathfrak{T}(T_1.T_2) \equiv \mathfrak{T}(T_1) \wedge \mathfrak{T}(T_2)$. Applying $\mathfrak{T}$ in this case, we get the following formula

$$\langle a, \texttt{:p}, b\rangle \in G \wedge \langle a, \texttt{:p}, \texttt{:n}\rangle \in G$$

To obtain the answer, i.e. all pairs a, b, we define a set comprehension of the formula over all its free variables

$$\{\langle a, b\rangle \,|\, \langle a, \texttt{:p}, b\rangle \in G \wedge \langle a, \texttt{:p}, \texttt{:n}\rangle \in G\}$$

Filter. The same approach can be elaborated to cover the core of the SPARQL pattern matching language. The first elaboration is that not all variables are free. Consider the slightly more complicated query in Fig. 2a that adds a filter that makes sure that an additional triple `?c :q ?b` exists in the dataset. In SPARQL 1.1, filters are expressions, and their definitions are in Fig. 4.



```
:a :q :m
:a :q :y
:x :p :y
:x :p :n
:y :r :z
:x :num 1
:y :num 1
```

(a) Data

```
SELECT *
WHERE {
  ?a :p ?b  T1
  ?a :p :n  T2
}
```

(b) Query

?a	?b
:x	:n
:x	:y

(c) Answer

Fig. 1. BGP example

```
SELECT * WHERE {  T1
  ?a :p ?b
  ?a :p :n
  FILTER EXISTS {  T2
   ?c :q ?b
 }}
```

(a) FILTER EXISTS

```
SELECT * WHERE {  T1
  ?a :p ?b
  ?a :p :n
  UNION {  T2
   ?c :q ?b
 }}
```

(b) UNION

```
SELECT * WHERE {  T1
  ?a :p ?b
  ?a :p :n
  OPTIONAL {  T2
   ?c :q ?b
 }}
```

(c) OPTIONAL example

Fig. 2. Examples

The variable ?c is not mentioned in the main pattern, so the formulation is a little different; there must be some value for c for any pair a, b in the solution. Specifically, in our example, because ?c does not even appear in the main pattern, the set of variables in the filter T_2 that are unbound in the main pattern T_1 is statically known, and consists only of the variable c. Hence, for simple cases where the set $\{v_1 \ldots v_k\}$ of variables in the filter expression T_2 that are unbound in the main pattern T_1 is statically known, we define $\mathfrak{T}(T_1 \exists\ T_2) \equiv \mathfrak{T}(T_1) \wedge \exists v_1 \ldots v_k \mathfrak{T}(T_2)$. However, we shall see later in this section that it gets more complex (due to the fact that variables in T_1 can be dynamically bound in general). For now, we get the following formula once we wrap it in a set comprehension over the free variables

$$\{\langle a, b\rangle \,|\langle a, \texttt{:p}, b\rangle \in G \wedge \langle a, \texttt{:p}, \texttt{:n}\rangle \in G \wedge \exists\ c\ \langle c, \texttt{:q}, b\rangle \in G\}$$

The FILTER NOT EXISTS construct is the same except the existential is negated.

So far, the queries have all had the same set of bound variables for all answers, but there are two SPARQL constructs that do not have this property: UNION and OPTIONAL.

Union. UNION has the expected meaning of combining two child patterns and hence denotes a logical or in our formalism; $\mathfrak{T}(T_1 \cup T_2) \equiv \mathfrak{T}(T_1) \vee \mathfrak{T}(T_2)$. However, UNION also illustrates one of the more thorny parts of SPARQL, which is that variables get bound dynamically. Specifically, in our example in Fig. 2b, if the left branch of the UNION produces mappings, then only ?a and ?b are bound, but ?c is not (similarly, the right branch binds only ?b and ?c). If both branches produce mappings, then a specific solution mapping μ can have either ?a and ?b bound, or ?b and ?c bound. Simply expressing union as an $\vee$ does not capture this; that formula would be $u = \{\langle a, b, c\rangle \,|((\langle a, \texttt{:p}, b\rangle \in G \wedge \langle a, \texttt{:p}, \texttt{:n}\rangle \in G) \vee \langle c, \texttt{:q}, b\rangle \in G)\}$. Some solutions for Fig. 2b bind only ?a and ?b; this corresponds to the left branch of the *or* in the set comprehension u; in that case there is no constraint on ?c and so every value for ?c would be a valid answer.

We need to capture in our formula that the left branch of the union binds only ?a and ?b; we introduce $dom(T)$ to denote variables bound in a solution. Thus, $dom\,(T1) = \{?\texttt{a}, ?\texttt{b}\}$ and $dom\,(T2) = \{?\texttt{b}, ?\texttt{c}\}$. Combining these for $dom(T)$ is slightly tricky: which other variable is bound depends on which side of the union was taken; we need to ensure that $dom(T)$ is either $dom(T_1)$ or $dom(T_2)$. A simple union will not suffice for this, so we use an auxiliary variable τ_i to denote the chosen branch:

$$dom(T) \equiv \exists \tau_i \begin{cases} dom(T1) \ \tau_i \\ dom(T2) \ \neg\tau_i \end{cases}$$

We also introduce a helper function $valid(x) \equiv \text{‘}\mathtt{x}' \in dom(T) \vee x = \emptyset$ to mean that a variable is either bound or is null; i.e. *valid* ensures that variables unbound in the pattern must be null rather than ranging over all possible values. We use `'x'` to denote the name of a variable, and we will use $valid(x_1, \ldots, v_n)$ to mean element-wise application of *valid*.

The formal definition of $dom(T)$ is given in Fig. 3. Given $dom(T)$, we get the following formula for this example. Hereinafter, we will implicitly add to every top level comprehension the constraint that all free variables are valid

$$\left\{ \langle a, b, c \rangle \,\middle|\, \left(\left(\begin{matrix} \langle a, \mathtt{:p}, b \rangle \in G \wedge \\ \langle a, \mathtt{:p}, \mathtt{:n} \rangle \in G \end{matrix} \right) \vee \langle c, \mathtt{:q}, b \rangle \in G \right) \wedge valid(a, b, c) \right\}$$

Optional. OPTIONAL patterns are optional in the sense that, if the pattern matches in the dataset, any additional variables bound in the optional pattern are bound in the overall pattern; if the optional pattern fails to match, those variables are left unbound. As shown in Fig. 2c, OPTIONAL patterns are also a complex kind of pattern since they generate two kinds of results: the first is when both the left and right hand sides are true, and the second is when the left hand side is true and the right hand side cannot be matched given the bound variables in the left hand side, which is essentially a filter not exists pattern

$$\mathfrak{T}(T_1 \text{ optional } \mathrm{T}_2) \equiv (\mathfrak{T}(\mathrm{T}_1) \wedge \mathfrak{T}(\mathrm{T}_2)) \vee \mathfrak{T}(\mathrm{T}_1 \nexists \mathrm{T}_2)$$

and

$$dom(T_1 \text{ optional } \mathrm{T}_2) \equiv \begin{cases} dom(T_1) \cup dom(T_2) & \mathfrak{T}(T_2) \\ dom(T_1) & otherwise \end{cases}$$

The complication with OPTIONAL is that to define the notion of the right hand side T_2 that cannot be matched given the bound variables in the left hand side T_1, we need to handle the dynamism of mappings of T_1, so for OPTIONALs, *dom* directly impacts the variables used in the existential quantification of subformulae. Hence, we define a helper logical operator for quantification, $\exists^T$. Intuitively, if a free variable v of $\mathfrak{T}(T_2)$ is bound in T_1, then $\exists^{T_1} v\, \mathfrak{T}(T_2)$ is simply $\mathfrak{T}(T_2)[v^{T_1}/v]$, the formula obtained after replacing all occurrences of v in $\mathfrak{T}(T_2)$ by v^{T_1} (where v^{T_1} refers to the constant to which v is bound in pattern T_1); otherwise, $\exists^{T_1} v\, \mathfrak{T}(T_2)$ corresponds to the normal existential quantification $\exists\, v\, \mathfrak{T}(T_2)$. This is precisely the behavior we want to account for the dynamism of mappings. We now formally define $\exists^T$ as follows: for a logical formula Q and a non-empty subset $\{v_1, \ldots, v_n\}$ of free variables of Q, $\exists^T v_1, \ldots, v_n Q$ means

$$\exists v_1, \ldots, v_n (\text{‘}\mathtt{v_1}\text{’} \in dom(T) \rightarrow v_1 = v_1^T) \wedge \ldots \wedge (\text{‘}\mathtt{v_n}\text{’} \in dom(T) \rightarrow v_n = v_n^T) \wedge Q$$

where v_i^T refers to the constant to which v_i is bound in pattern T. With a slight abuse of notation, when the set of variables is empty, we define, for a given formula Q, $\exists^T Q$ as $\exists^T Q \equiv Q$.

Going back to the FILTER EXISTS example in Fig. 2a, with the definition of $\exists^T$, the proper meaning of the FILTER EXISTS construct that accounts for

$$
\begin{aligned}
FVars(\mathfrak{T}(T)) &\equiv \text{ set of free variables of } \mathfrak{T}(T)\\
v^T &\equiv \text{ the constant to which the variable } v \text{ is bound in pattern } T\\
\exists^T\, v_1 \ldots v_k Q &\equiv \exists v_1 \ldots v_k \left(\text{`}\mathtt{v_1}\text{'} \in dom(T) \rightarrow v_1 = v_1^T\right) \wedge \ldots \wedge \left(\text{`}\mathtt{v_k}\text{'} \in dom(T) \rightarrow v_k = v_k^T\right) \wedge Q\\
\exists^T\, Q &\equiv Q\\
\nexists^T\, v_1 \ldots v_k Q &\equiv \neg(\exists^T\, v_1 \ldots v_k Q)
\end{aligned}
$$

$$
\mathfrak{T}(T,G) \equiv \begin{cases}
\langle a,b,c\rangle & \equiv \langle \mathfrak{G},a,b,c\rangle \in G\\
\text{VALUES } \mathrm{v}_1 \ldots \{\{\mathrm{c}_{1,1}\ldots\}\ldots\} & \equiv (v_1 = c_{1,1} \wedge \ldots) \vee \ldots\\
\text{BIND v AS } \mathrm{E}_1 & \equiv V(E_1,T) \wedge v = \mathfrak{E}(T,E_1)\\
s \text{ path o} & \equiv \langle s,o\rangle \in \mathfrak{P}(path)\\
\text{GRAPH g } \{\mathrm{T}_1\} & \equiv \text{let } \mathfrak{G} = \text{g in } \mathfrak{T}(\mathrm{T}_1)\\
T_1 .\, T_2 & \equiv \mathfrak{T}(T_1) \wedge \mathfrak{T}(T_2)\\
T_1 \text{ FILTER } \mathrm{E}_1 & \equiv \mathfrak{T}(T_1) \wedge V(E_1,T) \wedge \mathfrak{E}(T,E_1)\\
T_1 - T_2 & \equiv \begin{cases} \mathfrak{T}(T_1 \nexists T_2) & (dom(T_1) \cap dom(T_2)) \neq \emptyset\\ \mathfrak{T}(T_1) & otherwise \end{cases}\\
T_1 \cup\ T_2 & \equiv \mathfrak{T}(T_1) \vee \mathfrak{T}(T_2)\\
T_1 \text{ optional } \mathrm{T}_2 & \equiv (\mathfrak{T}(T_1) \wedge \mathfrak{T}(T_2)) \vee \mathfrak{T}(T_1 \nexists T_2)
\end{cases}
$$

$$
dom(T) \equiv \begin{cases}
\langle a,b,c\rangle & \equiv \text{names of all variables in the triple}\\
\text{VALUES } \mathrm{v}_1 \ldots \{\{\mathrm{c}_{1,1}\ldots\}\ldots\} & \equiv \{\text{`}v_1'\ldots\}\\
\text{BIND v AS e} & \equiv \{\text{`}v'\}\\
\text{GRAPH g } \{\mathrm{T}_1\} & \equiv \begin{cases} \{\text{`}g'\} \cup dom(T_1) & \text{g is a variable}\\ dom(T_1) & otherwise \end{cases}\\
T_1 .\, T_2 & \equiv dom(T_1) \cup dom(T_2)\\
T_1 \text{ FILTER } \mathrm{E}_1 & \equiv dom(T_1)\\
T_1 - T_2 & \equiv dom(T_1)\\
T_1 \cup T_2 & \equiv \begin{cases} dom(T_1) & \mathfrak{T}(T_1) \wedge \neg\mathfrak{T}(T_2)\\ dom(T_2) & \mathfrak{T}(T_2) \wedge \neg\mathfrak{T}(T_1)\\ \exists \tau_i \in \{\text{true, false}\} \text{ such that} & \\ \text{if } \tau_\text{i} \text{ dom}(\mathrm{T}_1) \text{ else dom}(\mathrm{T}_2) & \mathfrak{T}(T_1) \wedge \mathfrak{T}(T_2) \end{cases}\\
T_1 \text{ optional } \mathrm{T}_2 & \equiv \begin{cases} dom(T_1) \cup dom(T_2) & \mathfrak{T}(T_2)\\ dom(T_1) & \neg\mathfrak{T}(T_2) \end{cases}
\end{cases}
$$

$$
\mathfrak{P}(p) \equiv \left\{
\begin{aligned}
\mathfrak{P}(link:p) &\equiv \{< x,y > | \langle \mathfrak{G}, x, :p, y\rangle \in G\}\\
\mathfrak{P}(inv:p) &\equiv \{< x,y > | \langle \mathfrak{G}, y, :p, x\rangle \in G\}\\
\mathfrak{P}(NPS(p_1 \ldots p_n)) &\equiv \{< x,y > | \exists q\ q \notin p_1 \ldots p_n \wedge \langle \mathfrak{G}, x, q, y\rangle \in G\}\\
\mathfrak{P}(seq(p_1,p_2)) &\equiv (\mathfrak{P}(p_1).\mathfrak{P}(p_2))\\
\mathfrak{P}(alt(p_1,p_2)) &\equiv (\mathfrak{P}(p_1) \cup \mathfrak{P}(p_2))\\
\mathfrak{P}(OneOrMorePath(p_1)) &\equiv \hat{}\,\mathfrak{P}(p_1)\\
\mathfrak{P}(ZeroOrOnePath(p_1)) &\equiv \mathfrak{P}(p_1) \cup \{\langle x,x\rangle | \exists y,z\ \langle \mathfrak{G},x,y,z\rangle \in G \vee \langle \mathfrak{G},y,z,x\rangle \in G\}\\
\mathfrak{P}(ZeroOrMorePath(p_1)) &\equiv \mathfrak{P}(alt(OneOrMorePath(p_1), ZeroOrOnePath(p_1)))
\end{aligned}
\right.
$$

Fig. 3. Pattern semantics. (We use $T_1 \nexists T_2$ to mean T_1 FILTER NOT EXISTS T_2). Note that this handles *named graphs*, which essentially generalize triples to quads. For ease of explanation, our examples just use triples.

the dynamism of mappings is as follows: $\mathfrak{T}(T_1 \exists T_2) \equiv \mathfrak{T}(T_1) \wedge \exists^{T_1} v_1 ... v_k \mathfrak{T}(T_2)$ where $\{v_1 ... v_k\}$ is the set of free variables of $\mathfrak{T}(T_2)$.

Now, we can define $\nexists^T$ as follows: for a logical formula Q and a subset $S = \{v_1, \ldots, v_n\}$ of free variables of Q (S may be empty), $\nexists^T v_1, \ldots, v_n Q \equiv \neg(\exists^T v_1, \ldots, v_n Q)$. Finally, we can formally define the FILTER NOT EXISTS construct: $\mathfrak{T}(T_1 \nexists T_2) \equiv \mathfrak{T}(T_1) \wedge \nexists^{T_1} v_1 ... v_k \mathfrak{T}(T_2)$ where $\{v_1 ... v_k\}$ is the set of free variables of $\mathfrak{T}(T_2)$.

Minus. MINUS is essentially the same as filter not exists, but it adds the constraint that the left and right hand sides must have at least one variable in common in order for subtraction to occur. That is captured by adding a *dom* constraint to filter not exists:

$$\mathfrak{T}(T_1 - T_2) \equiv \begin{cases} \mathfrak{T}(T_1 \nexists T_2) & (dom(T_1) \cap dom(T_2)) \neq \emptyset \\ \mathfrak{T}(T_1) & otherwise \end{cases}$$

Property Paths. A significant new feature in SPARQL 1.1 is property paths, which essentially allow specification of a regular expression over predicates to connect a given subject and object. Relational logic is natural to define such predicates as relations. We define property paths as $\mathfrak{P}$ in Fig. 3.

Aggregation. Aggregation is specified over a set comprehension, as defined in Fig. 4; We illustrate with the query `select ?a (max(?n) as ?m) where ?a :q ?b . ?b :num ?n GROUP BY ?a`. The group is specified in this case as those answers sharing the same value of ?a given a formula P:

$$G(P) = (\text{let } \mathrm{a_g} = \mathrm{a} \text{ in } \{\langle \mathrm{a}, \mathrm{b}, \mathrm{n} \rangle \,|\, \mathfrak{T}(\mathrm{P}) \wedge \mathrm{a} = \mathrm{a_g}\})$$

Then the aggregation function is applied; first the expression being aggregated is evaluated for each group member,

$$E(G) = \{n \,|\, \langle a, b, n \rangle \in G\}$$

and then the aggregate function itself is computed:

$$max(E) \in E \wedge \nexists max' \in E \wedge max' > max(E)$$

1.2 Expressions

Expressions appear in filter and bind patterns, and represent values computed from other variables and constants. Expression semantics $\mathfrak{E}$ is presented in Fig. 4. An example using bind is Figure ??. This expression appears normal, as it simply adds two variables together.

Validity means that evaluation is allowed only when every portion of the expression is valid, i.e. applicable to its arguments, if any. It is analogous to type correctness in dynamic languages, but tests simply fail when they are invalid rather than aborting the computation with a type error. Note that && and || operations swallow errors when possible. && returns false if either argument is valid and returns false, regardless of whether the other argument is valid. || is analogous.

$$\mathfrak{E}(E,T) \equiv \begin{cases} E_1 == E_2 \equiv & \begin{array}{ll} \multicolumn{2}{l}{\mathrm{let\ v_1 \leftarrow \mathfrak{E}(E_1,T), v_2 \leftarrow \mathfrak{E}(E_2,T)\ in}} \\ v_1 == v_2 & B(T(v_1)) \land B(T(v_2)) \\ S(v_1) == S(v_2) \land T(v_1) == T(v_2) & otherwise \end{array} \\ E_1 \neq E_2 \equiv & \begin{array}{ll} \multicolumn{2}{l}{\mathrm{let\ v_1 \leftarrow \mathfrak{E}(E_1,T), v_2 \leftarrow \mathfrak{E}(E_2,T)\ in}} \\ v_1 \neq v_2 & B(T(v_1)) \land B(T(v_2))_2 \\ false & otherwise \end{array} \\ \neg E_1 \equiv & \neg \mathfrak{E}(E_1,T) \\ E_1 \ \&\& \ E_2 \equiv & \mathfrak{E}(E_1,T) \land \mathfrak{E}(E_2,T) \\ E_1 \parallel E_2 \equiv & \mathfrak{E}(E_1,T) \lor \mathfrak{E}(E_2,T) \\ bound(v) \equiv & 'v' \in dom(T) \\ \mathrm{EXISTS\ T_1} \equiv & \exists^T v_1 \ldots v_k \ \mathfrak{T}(T_1) \ \text{ with } v_i \in FVars(\mathfrak{T}(T_1)) \\ \mathrm{NOT\ EXISTS\ T_1} \equiv & \nexists^T v_1 \ldots v_k \ \mathfrak{T}(T_1) \ \text{ with } v_i \in FVars(\mathfrak{T}(T_1)) \end{cases}$$

$$V(E,T) \equiv \begin{cases} E_1 == E_2 \equiv V(E_1,T) \land V(E_2,T) \\ E_1 \neq E_2 \equiv V(E_1,T) \land V(E_2,T) \\ \neg E_1 \equiv V(E_1,T) \\ E_1 \ \&\& \ E_2 \equiv (V(E_1,T) \land V(E_2,T)) \lor (V(E_1,T) \implies \neg\mathfrak{E}(E_1,T)) \lor (V(E_2,T) \implies \neg\mathfrak{E}(E_2,T)) \\ E_1 \parallel E_2 \equiv (V(E_1,T) \land V(E_2,T)) \lor (V(E_1,T) \implies \mathfrak{E}(E_1,T)) \lor (V(E_2,T) \implies \mathfrak{E}(E_2,T)) \\ otherwise \quad true \end{cases}$$

$$\mathfrak{G}(T, e_{1\ldots n}) \equiv \mathrm{let\ e_1^{'} = e_1, \ldots e_n^{'} = e_n in} \left\{ \mathrm{FVars(\mathfrak{T}(T))} \middle| \mathrm{\mathfrak{T}(T) \land e_1^{'} = e_1 \land \ldots e_n^{'} = e_n} \right\}$$

$$\mathfrak{A}(T, a_{1\ldots m}, e_{1\ldots n}) \equiv \mathfrak{T}(T) \land var(a_1) = agg(a_1) \left(\{ exp(a_1)(t) \, | t \in \mathfrak{G}(T, e_{1\ldots n}) \} \right) \land \ldots$$

$$T(v) \equiv \text{RDF type of value } v$$

$$S(v) \equiv \text{literal string of value } v$$

$$B(t) \equiv \text{true if type } t \text{ is built in}$$

Fig. 4. Expression semantics

1.3 Blank Node Equivalence

Entailment regimes [5] define how SPARQL query answers change for RDF graphs under a range of OWL [7,8] semantics. We cover RDF entailment, under which two RDF graphs G_1 and G_2 are equivalent if they differ only in the labels of their blank nodes, denoted $G_1 =_{RDF} G_2$.

2 Conclusion

To our knowledge, we present the most complete semantics for SPARQL that is not based on translation or has actually been run. Previous work based on directly assigning meaning to SPARQL constructs [1] focuses on a limited subset of SPARQL. Other formalisms for SPARQL have been defined based on translations to Answer Set Programming (ASP) [9], Datalog [11] and relational algebra [2]. None of these have been subject to automated verification.

We presented a formal definition of the core of SPARQL 1.1 that parsimoniously distills the language down to a few constructs, succinctly capturing subtle issues like the meanings of negation. This semantics is expressed in relational logic, and has been mechanically checked by implementing it.

References

1. Arenas, M., Pérez, J.: Querying semantic web data with SPARQL. In: Lenzerini, M., Schwentick, T. (eds.) Proceedings of 30th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2011, Athens, Greece, 12–16 June 2011, pp. 305–316. ACM (2011). http://doi.acm.org/10.1145/1989284.1989312
2. Cyganiak, R.: A relational algebra for SPARQL. Technica report, Digital Media Systems Laboratory, HP Laboratories Bristol (2005)
3. data.gov. http://data.gov/
4. DBPedia SPARQL Endpoint. http://dbpedia.org/sparql
5. Sprql 1.1 entailment regimes. http://www.w3.org/TR/sparql11-entailment/
6. GeoNames Semantic Web. http://datahub.io/dataset/geonames-semantic-web
7. Owl 2 web ontology language document overview. http://www.w3.org/TR/owl2-overview/
8. Owl 2 web ontology language mapping to RDF graphs. www.w3.org/TR/owl2-mapping-to-rdf/
9. Polleres, A., Wallner, J.P.: On the relation between SPARQL 1.1 and answer set programming. J. Appl. Non-Class. Log. **23**(1–2), 159–212 (2013)
10. Prud'hommeaux, E., Seaborne, A.: SPARQL query language for RDF, January 2008. http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/
11. Schenk, S.: A SPARQL semantics based on datalog. In: Hertzberg, J., Beetz, M., Englert, R. (eds.) KI 2007. LNCS (LNAI), vol. 4667, pp. 160–174. Springer, Heidelberg (2007)
12. Semantic web use cases. http://www.w3.org/2001/sw/sweo/public/UseCases/
13. Semantic XBRL. http://datahub.io/package/semantic-xbrl
14. UniProt SPARQL Endpoint. http://beta.sparql.uniprot.org

Non-traditional Environments

Towards a Scalable Framework for Artifact-Centric Business Process Management Systems

Jiankun Lei, Rufan Bai, Lipeng Guo, and Liang Zhang(✉)

School of Computer Science, Shanghai Key Laboratory of Data Science,
Fudan University, Shanghai, China
{jklei,rfbai,lpguo,lzhang}@fudan.edu.cn

Abstract. Over the last decade, we have witnessed the success of artifact-centric approach in business process management (BPM). However, the scalability issue was neglected by almost all its implementation frameworks. A non-scalable framework will severely hamper applicability of corresponding artifact-centric BPM systems in large-scale applications. Considering distinct characteristics of the Representational State Transfer (REST) architectural style, we propose a distributed artifact-centric BPM framework based on REST principles. A prototype is developed using Docker-based micro services for continuously delivering of process engine instances. Through extensive experiments against a typical process-aware application, we confirm that the proposed framework is promising to support scalable artifact-centric BPM systems.

Keywords: Artifact-centric business process · REST · Scalable framework

1 Introduction

Over the last decade, the artifact-centric approach [4,13] has emerged as a popular perspective in business process management (BPM). Comparing to traditional activity-centric BPM, the artifact-centric BPM approach focuses on data with lifecycle, known as *artifacts*, in business processes and provides a better balance between process-centric and data-centric modeling [20]. As it rapidly develops, more and more artifact-centric BPM systems are deployed to support enterprises' operational tasks, the systems inevitably have to handle a large number of users and business process instances in some process-aware applications. This tendency requires corresponding infrastructures, e.g. the frameworks and implemented process engines, to be scalable enough to meet the challenge.

Unfortunately, due to its infancy, almost all the implementation frameworks of the artifact-centric BPM, ranging from ArtiFlow [10] (and its successor EZ-Flow [3]), to ACP [12], and Barcelona [7], have not yet addressed the scalability issue properly. Taking the ACP framework [12] as an example, the process engine exploits a centralized rule evaluation facility (i.e. Drools) to actuate services which have artifacts transferred along their lifecycles. The centralized nature of the rule engine prevents ACP from being heavily-distributed deployed,

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 309–323, 2016.
DOI: 10.1007/978-3-319-48743-4_25

which severely hurts the scalability. Our previous studies focus on the realization of artifact-centric workflows [10] or a workflow engine [3], had no concerns about scalability, too. From our experiments, we can find that Barcelona behaves poorly in terms of scalability as well.

Speaking of scalability, it is widely acknowledged that the World Wide Web (WWW) is a successful, distributed, and scalable platform [6]. The success is built so much on the underlying Representational State Transfer (REST) architectural style. REST provides principles behind the WWW, which is responsible for many desirable properties, such as scalability, modifiability, and interoperability [5,18]. We believe that it would be promising to introduce the REST style to artifact-centric BPM systems for the purpose of scalability.

In fact, there are some studies in the area of the RESTful business process modeling. Kumaran *et al.* [9] has proposed a RESTful architecture for business processes based on business entities modeled as Mealy machines. Other researches [14,21] take the resource-oriented approach of REST for activity-centric business processes to improve the simplicity of interfaces and interoperability of services. However, on one hand, approaches like [14,21], cannot take advantage of the artifact-centric approach, and on the other hand, the three studies all lack implementation or experiments. As a result, we cannot confirm the scalability of these approaches.

To bridge the gap between the study of artifact-centric BPM and scalable BPM framework, we propose a distributed artifact-centric BPM framework, named ArtiREST, based on REST principles. We also develop a prototype by using Docker-based micro services for continuously delivering of process engine instances. Extensive experiments towards the scalability demonstrate the superiority of the framework over other available ones.

The technical contribution of this paper can be summarized as:

– Identifying the scalability issue in artifact-centric BPM;
– Proposing a distributed artifact-centric BPM framework leveraged by the REST style and developing a prototype to prove the feasibility of the framework, and
– Confirming the applicability of REST style in artifact-centric BPM in pursuit of better scalability.

The remainder of this paper is organized as follows. Section 2 presents a running example to illustrate the proposed framework in subsequent sections. Section 3 specifies the scalability from many dimensions. Section 4 presents the ArtiREST in detail, and Sect. 5 evaluates framework in terms of scalability by experiments. Section 6 compares our approach with some related work. Finally, we give the conclusion and our future work.

2 A Running Example

Artifacts are business-relevant entities that are created and evolved through business processes. An artifact class (hereafter artifact) defines an information model and its lifecycle [4]. Consider a traditional loan approval process [16] (the middle part), and its artifact-centric representation (the upper part) in Fig. 1. The later defines an artifact *Loan*, which consists of an information model (right part) and a lifecycle (upper left). There are six states in the lifecycle (e.g. *Pending*) and five external services in the process (e.g. *Create* on the bottom). A client can instantiate the process by creating a loan artifact instance, and follows the process to choose a loan type to apply. A manager can approve or cancel the loan application. If it is approved and confirmed by the bank business services, current loan rate will be assigned to the *rate* attribute of the *Loan* artifact instance. Business rules (shaded diamonds) and services (rounded rectangles) work together to update attributes in the information model and have the artifact instance traveled from one state to another along its lifecycle.

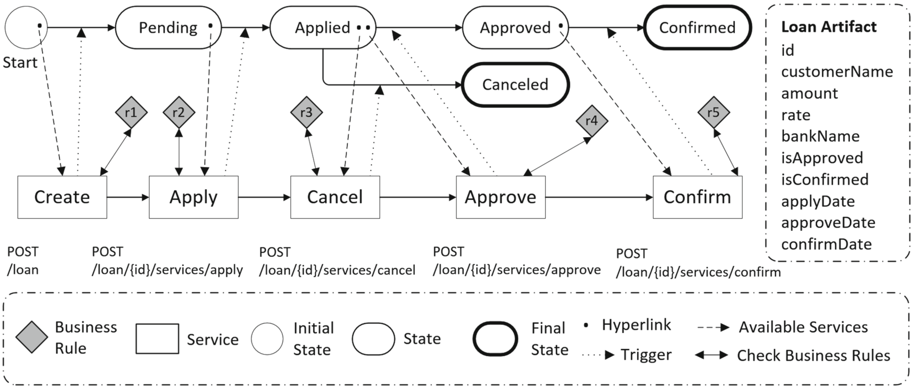


Fig. 1. An artifact-based view of a loan approval process

One of the significant innovations of artifact-centric BPM is that it focuses on what can be done (goals and progresses) instead of what should be done (tasks or corresponding services) [1]. Nevertheless, the advantages of artifact-centric BPM rely on the scalability of the underlying framework. Before discussing our framework, it is the time to elaborate on the scalability first.

3 Scalability

Scalability is a desirable property of a system, especially in the case as its load increases. Usually, scalability refers to the capability of a system, network, or process to handle a growing amount of work, or its potential to be enlarged in order to accommodate the growth [2]. In our context, we define scalability as,

Definition 1 (Scalability). A system is scalable if it can provide services and varies the quality of services (QoS) gracefully under light, moderate, to heavy loads.

The QoS can be measured from different angles. In this study, we are interested in performance. Hence, we choose the transaction per second (TPS) and the average response time (ART) as two indictors of QoS. As we know, investing more resources in a system can improve its QoS to some extent. Methods of adding more resources to a system fall into two broad categories: horizontal and vertical scaling.

Definition 2 (Horizontal Scalability). A system is horizontally scalable (or scale out/in) if it is scalable as adding more nodes to (or removing nodes from) the system.

Since Barcelona [7] and ACP [12] frameworks are implemented on a single server rather than a distributed platform, we need to consider vertical scalability as well.

Definition 3 (Vertical Scalability). A system is vertically scalable (or scale up/down) if it is scalable as adding resources to (or removing resources from) a single node in the system.

Technically, scalability improvement strategies include changing scheduling rules over shared resources, avoiding centralized control, or leveraging parallelism. We adopt these strategies to develop a distributed framework for artifact-centric BPM systems.

4 ArtiREST BPM Framework

In this section, we present the artifact-centric business process model, and elaborate its RESTful representation, which results in the ArtiREST, a distributed framework for artifact-centric BPM systems.

4.1 Artifact-Centric Business Process Modeling

There are plethoric studies to specify artifact-centric business process models, formally or informally. But they reach a consensus that such a business process is dominated by business artifacts.

In this study, we adopt the ACP model [12] with some modifications. Specifically, an artifact class is a tuple (A, Q, s, F), where A is a finite set of attributes, Q is a finite set of states, $s \in Q$ denotes the pseudo initial state and $F \subseteq Q$ is a set of final states. Let Z be a schema of artifacts, which is a finite set of artifact classes. A business rule r is a triple (λ, β, v) where λ is pre-condition and β is post-condition. Table 1 shows a business rule of our running example.

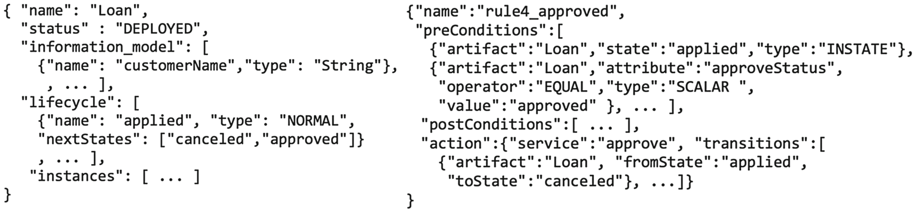
An artifact instance is actuated along its lifecycle by services constrained by business rules. Without loss of generality, we define services as RESTful web services [15].

Table 1. An example of business rule in the loan approval process

Post-condition	$instate(L, Pending)$
r2: Apply for loan L	
Pre-condition	$instate(L, Pending) \wedge defined(L, amount) \wedge defined(L, bankName)$ $\wedge\ amount > 10000$
Service	$apply(L)$
Post-condition	$instate(L, Applied)$

Definition 4 (Service). A service over a schema Z is a tuple (n, u, m, V_r, V_w), where n is a unique service name, u the URI and m a REST method (e.g. GET, PUT, POST, DELETE, etc. in the context of HTTP). V_r and V_w are finite sets of variables of artifact classes in Z.

As illustrated in Fig. 1, each state contains links (dots) to available services that might make state transitions. In this sense, the model conforms to the famous HATEOAS principle of REST [6]. It is worth emphasizing the importance of HATEOAS in our framework. Relying on the principle, we reverse the relation of business rules and services in traditional ACP model, which relieves us from the centralized control. A process engine needs only to evaluate those business rules (shaded diamonds) bound to a service (rounded rectangle) locally, instead of all rules globally when receiving an event like ACP. For example, if it is going to invoke service *apply*, the engine now only evaluates business rule $r2$. This modification to ACP can improve scalability drastically. In our modified ACP model, business rules are guards to guarantee the correct invocation of services and the proper transition among states. A service invocation is correct only if its pre-conditions hold before invoking the service and the post-conditions hold after the service has been invoked. A proper transition between two states happens only if the post-condition of the business rule stands.



```
{ "name": "Loan",
  "status" : "DEPLOYED",
  "information_model": [
    {"name": "customerName","type": "String"},
     , ... ],
  "lifecycle": [
    {"name": "applied", "type": "NORMAL",
    "nextStates": ["canceled","approved"]}
    , ... ],
   "instances": [ ... ]
}
```

```
{"name":"rule4_approved",
 "preConditions":[
   {"artifact":"Loan","state":"applied","type":"INSTATE"},
   {"artifact":"Loan","attribute":"approveStatus",
    "operator":"EQUAL","type":"SCALAR ",
    "value":"approved" }, ... ],
  "postConditions":[ ... ],
  "action":{"service":"approve", "transitions":[
    {"artifact":"Loan", "fromState":"applied",
     "toState":"canceled"}, ...]}
}
```

Fig. 2. A JSON representation of artifact classes *Loan* in resource `/{process}/{process_id}/artifacts/{artifact}` (left); A JSON representation of business rules *r4* in resource `/{process}/{process_id}/rules/{rule}`

4.2 RESTful Business Processes

The core abstraction of information in RESTful systems is a resource with a URI [6]. In ArtiREST, we use templates in Table 2 to represent resources related to an artifact-centric business process at the different levels of granularity. In addition to them, we can zoom in on the lifecycle of an artifact as Fig. 2 (left) or on the business rules affiliated to it as Fig. 2 (right).

Both of business information model and its lifecycle of an artifact can be easily organized as web resources. RESTful services and business rules can be also represented as resources. Back to our running example, `/loan/1/services/apply` represents the URI of the service *apply* in Fig. 1,

Table 2. Uniform interface semantics for process-related resources

Method	Description
`Process:/{process}`	
`GET`	Retrieve a representation of the process, with links to its instances
`PUT`	Deploy or update the process
`DELETE`	Undeploy the process
`POST`	Create a new process instance, which may instantiate some artifact intances
`Process instance:/{process}/{process_id}`	
`GET`	Retrieve a representation of the process instance. With links to its artifact instances
`DELETE`	Delete the process instance
`Artifact:/{process}/{process_id}/artifacts/{artifact}`	
`GET`	Retrieve a representation of an artifact, with links to its instances
`POST`	Instantiate a new artifact instance
`Artifact instance:/{process}/{process_id}/artifacts/{artifact}/{instance}`	
`GET`	Retrieve the information of the artifact instance, including current state, attributes, state transitions and logs
`PUT`	Update the attributes of the instance
`DELETE`	Delete the artifact instance
`Service:/{process}/{process_id}/services/{service}`	
`GET`	Retrieve a representation of the service
`PUT`	Bind or update the service
`DELETE`	Unbind or delete the service
`POST`	Invoke the service initiatively
`Business rule:/{process}/{process_id}/rules/{rule}`	
`GET`	Retrieve a representation of the business rule, with its pre-condition, post-condition, corresponding service and etc.
`PUT`	Update the business rule
`DELETE`	Delete the business rule

where `1` is the process instance id and *apply* is the service name. To invoke a service, we apply standard HTTP verbs on its URI. For example, a client can simply invoke the interface `POST /loan/1/services/apply` with proper parameters to apply for a new loan.

4.3 Design Issues

REST is an architectural style for building Internet-scale distributed hypermedia systems [6]. Its scalability comes from its style principles, such as global addressability, client-server, stateless interaction, uniform interface, and layered system. ArtiREST inherits REST's desirable architecture properties, e.g. performance, scalability, simplicity, modifiability, and interoperability by enforcing these principle constraints in both the framework and its system architecture. A system architecture conforms to ArtiREST framework is shown in Fig. 3.

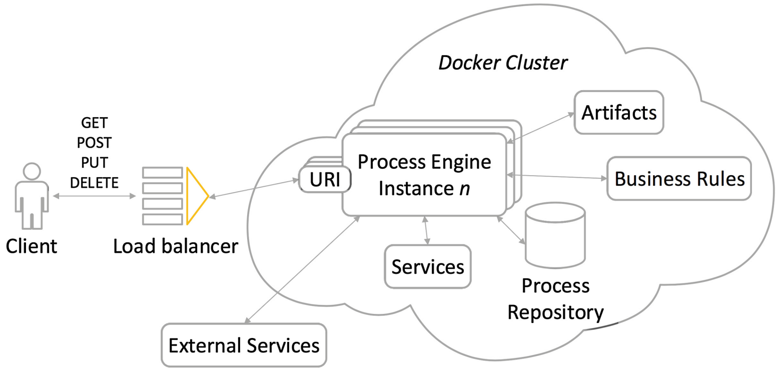


Fig. 3. A distributed architecture for ArtiREST system

- **Layered System.** The architecture is organized as multiple layers, including proxy, backend servers, cache, and database. The proxy layer provides load balancing of services across multiple networks and backend servers.
- **Stateless.** Each request from a client contains all information necessary for the server to understand and no client context is stored on the server between requests [6].
- **Uniform Interface.** A client can only make requests through HTTP verbs, i.e. GET, PUT, POST, DELETE, etc.
- **Cacheable.** The architecture employs multiple layers of cache, such as proxy cache and database cache.

We implement a light-weight and web-based BPM prototype, also named ArtiREST. Comparing with the existing artifact-centric BPM systems, we have made few decisions on some design issues.

- **Distributed Process Engines.** Like ACP [12], ArtiREST supports automated realization of artifact-centric models under one prerequisite: all services have been implemented, registered, and deployed. It is practical because the service-oriented approach promotes services registration. Unlike ACP and other centralized control implementations, ArtiREST advocates distributed evaluation by incorporating REST's HATEOAS and micro services. Each process engine is capable of evaluating those rules related to particular service independently. New engine instance can be instantiated on demand.
- **Cluster.** We deploy our prototype in a Docker Swarm Cluster, which enables booting a large number of containers in a very short time. Each container is built with all system services and is capable of serving all requests from clients. As the number of request increases, we can quickly scale up the container or scale out containers to improve the QoS of the system.
- **Sessions.** Session management in a distributed system is tough because of synchronization across servers. However, there should be no session according to the stateless principle of REST. One trick we are using is the token-based authentication [17], instead of keeping user sessions. After logging in, a unique token will be generated for the user. This mechanism requires that the HTTP header of any requests must contain the token for authentication.
- **Instance synchronization.** To further improve performance and scalability, we don't save process instances back to process repository once they are updated. Instead, we keep them in cache temporarily. Two mechanisms can be used to synchronize data between cache and backend repository: scheduled or fixed-point synchronization. It is controlled by policies that are application-sensitive and could be leveraged by a policy manager.
- **Data consistency.** The data in the database is consistent, but it may not be true in caches. Inconsistency frequently happens since the number of containers may vary with the current loads and some containers may crush sometimes. Many cache algorithms are proposed to ensure data consistency in the industry so we can use them directly.

4.4 Improving Scalability

Considering our model and system framework design, we address the following essential points for improving scalability specifically.

- **Decentralized business rules evaluation.** As mentioned in Sect. 4.1, we have well-defined REST services bound with related business rules. According to HATEOAS principle, a process instance regardless of any state only has limited available services. A user must explicitly invoke a specific service with self-describing messages, so the business engine only needs to evaluate the bound business rules.
- **RESTful architecture.** According to REST principles [6], *layered system*, *stateless*, *cacheable* mainly contributes to the system scalability. Load balancing can be performed among process engine instances in a distributed Docker

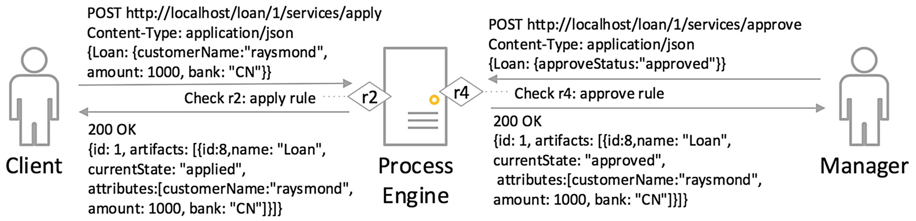
Swarm cluster. No centralized session (*stateless*) also improves scalability. Caches of business process models and instances on the server side further improve scalability.

4.5 BPM with ArtiREST

The concepts and technologies relevant to BPM can be outlined as a business process lifecycle: *design* & *analysis*, *configuration*, *enactment*, and *evaluation*, in a cyclical structure with logical dependencies [19]. In this section, we discuss how to support BPM with the ArtiREST prototype.

- **Design** & **Analysis.** We implemented a web-based graphic tool to support business process modeling. The tool is used to define artifacts, business rules, and service interfaces. Since the design model is serialized and saved in a standard format (XML/JSON), model validation can be done through the validation checking of XML or JSON. Verification of instances is supported as the execution model reflects the conceptual model according to ACP claims [12]. So far simulation is not automated, human participation must be involved to write or execute simulation cases.
- **Configuration.** Configuration becomes easier since we ship our services into Docker containers, which facilitates the system configuration and startup.
- **Enactment.** Process instantiation, service invocation and instance monitoring are involved in the enactment phase. According to the artifact-centric approach, the instantiation of artifact classes represents the process instantiation. Artifact instances can be serialized for persistence and transmission. Similarly, processes instance monitoring can be made via monitoring the artifact instances. The creation, update, or state transitions of an artifact will be logged, which provides information for monitoring. So far artifacts monitoring and logging have been implemented in our prototype while performance monitoring is absent. Process execution is accomplished by request resources and invoking RESTful web services.

  Taking the loan approval process in Fig. 1 as an example, the execution procedure is shown in Fig. 4. A client applies for a loan by invoking a service named *Apply*. The POST request is submitted to invoke the service, and the server will get the corresponding business rule *r2* and check whether the state and attributes of the artifact *Loan* satisfy the pre-conditions. If it is satisfied, the Process Engine will invoke the service and then check if the new state of the



```
POST http://localhost/loan/1/services/apply
Content-Type: application/json
{Loan: {customerName:"raysmond",
amount: 1000, bank: "CN"}}
```

```
200 OK
{id: 1, artifacts: [{id:8,name: "Loan",
currentState: "applied",
attributes:[customerName:"raysmond",
amount: 1000, bank: "CN"]}]}
```

```
POST http://localhost/loan/1/services/approve
Content-Type: application/json
{Loan: {approveStatus:"approved"}}
```

```
200 OK
{id: 1, artifacts: [{id:8,name: "Loan",
currentState: "approved",
 attributes:[customerName:"raysmond",
amount: 1000, bank: "CN"]}]}
```

Fig. 4. The enactment of a process of loan approval in ArtiREST

artifact satisfied the post-condition. If everything is successful, the server will response to the client with the applied *Loan* artifact. Otherwise, an alarm is pending. Similarly, a manager can approve the loan application via a POST request to invoke service *Approve*.
- **Evaluation.** Evaluation involves improvement on process models and implementations by some techniques, such as monitoring and process mining on instance logs. Thanks to the artifact-centric approach, it is easier to monitor artifacts than tracking a traditional activity-centric process instance.

5 Evaluation

The technical evaluation towards the artifact-centric business process modeling and realization were given in ACP framework [12]. We dont repeat the evaluation here but focus ourselves on scalability. We are going to test if ArtiREST can provide services and varies QoS gracefully under light, moderate, to heavy loads. We also compare the scalability of our prototype with the two famous implementations of artifact-centric BPM tools Barcelona [7] and ACP [12]. All experiments are conducted with the loan approval process as shown in Fig. 1.

5.1 Load Testing

Load testing makes it practical to measure a systems QoS by simulating user behavior [11]. There are two primary QoS factors we are interested: availability and response time. In the following experiments, we ensure services to be always available and focus on the response time and transactions per second.

The systems to be tested are deployed in a Docker Swarm cluster with ten physical nodes, including seven process engine nodes, one load balancer, one cache node, and one database node. Each physical node has 32 GB RAM and four cores with hyper-threading. So we treat each node as a computer with 8 (logical) CPUs in following experiments.

We are interested if we can use low-cost (limited resources) process engines to handle heavy loads. As the QoS drops, we will supply more engines to the system. In other words, we care about the horizontal scalability. To do so, we ship an engine into a Docker container with only 1 GB RAM and 1 CPU.

We use Apache JMeter[1] to simulate virtual users and monitor performance. Apache JMeter is convenient to generate heavy load with many server protocol types, such as HTTP, REST, SOAP, etc. We use it to simulate that a growing number, for example from 0 to 2000 in 45 s (ramp-up period), of virtual users invoke 5 RESTful web services to complete a loan approval process in Fig. 1. Theoretically, we want to simulate a linear increasing of load to find out the maximum number of users (capability) that a system can serve simultaneously.

Figure 5 shows the result of partial load tests. There are six subgraphs, each of them represents the performance curves with a certain number (1–6) of running process engine instances. For example, the first graph in Fig. 5 shows the

[1] Apache JMeter: http://jmeter.apache.org.

performance delivered by one process engine. The TPS reaches to a maximum value (about 40) before the response time begins to increase sharply. We call it a saturation point, usually happens when resources are exhausted in a system.

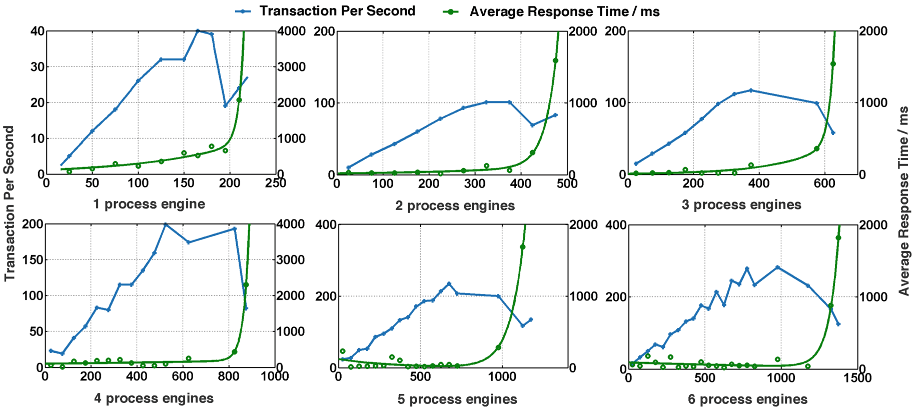


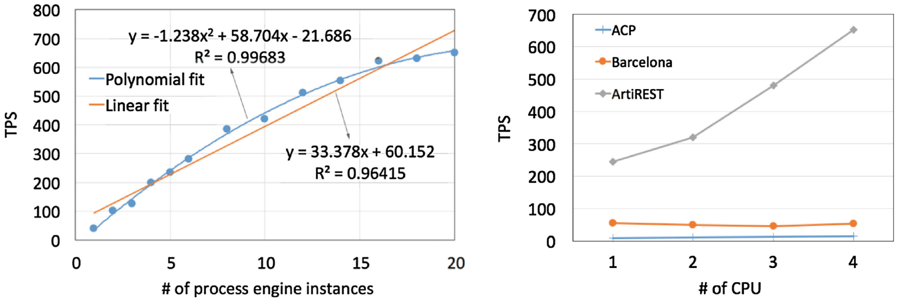
Fig. 5. Performance versus number of users. Here we observe two key performance metrics, i.e. the ART (average response time) on the left vertical axis, and TPS (transactions per second) on the right vertical axis. The annotation "3 process engine" indicates there are 3 active process engine instances.

5.2 Horizontal Scalability Evaluation

Usually, horizontal scalability can be measured by increasing loads steadily until the SLA (Service-Level Agreement) is achieved or the target resource utilization is reached. Resource utilization of distributed containers is difficult to measure, so we calculate the maximum TPS before saturation point as the SLA indicator.

Figure 5 shows that ArtiREST behaves well regarding horizontal scalability. As the increase of the number of process engine instances, the TPS rises steadily, from 40 up to 250, and supports more active users, from 150 up to 1000. Before saturation points, the response time is usually lower than 1 s.

Figure 6(a) shows the average number of maximum TPS before the saturation points vary with the number of process engines. In the linear fit, the equation is $y = 33.378x + 60.152, R^2 = 0.96415$, where y is an average number of the maximum TPS and x is the number of containers in the cluster. In the polynomial fit, the equation is $y = -1.238x^2 + 58.704x - 21.686, R^2 = 0.99683$. We can see that after the number of process engine instances exceeds 15, the TPS increases slowly. It is a normal phenomenon since we have only 10 physical nodes in the cluster, including process engines, cache, database, and load balancer. After the number of process engines exceeds a certain number, the cluster begins to saturate because of the limited resources for networks, database, cache, etc. We believe that the TPS can increase much faster and higher by adding more machines in the cluster and not restricting the RAM volume of each engine.



(a) Horizontal scalability: maximum TPS versus the number of process engines

(b) Vertical scalability: maximum TPS versus the number CPU

Fig. 6. Result of horizontal scalability and vertical scalability

5.3 Vertical Scalability Evaluation

The two mainstream implementations of artifact-centric BPM frameworks, Barcelona [7] and ACP [12], cannot scale out. Hence, we compare them with ArtiREST to investigate the vertical scalability by increasing the number of CPU in a single machine. To do it, we designed models for the loan process in accordance with ACP and Barcelona languages, respectively.

Figure 6(b) reveals the fact that ArtiREST scales vertically much better than the two competitors. By analysis, we know the huge difference results from many factors. On one hand, traditional frameworks focus on the realization of artifact-centric models but neglect the scalability. On the other hand, there are some flaws in design and implementation. For example, from the point view of design, ACP heavily relies on the business rule engine, in exchange for the flexibility processes modeling, it has to pay for the cost of centralized control. Consider the implementation techniques, there only exists a single thread in ACP framework as the centralized rule engine, and each business process instance is also realized as a single thread. Both the poor design and implementation flaws seriously harm its performance.

It should be noted that the maximum TPS with one business engine of ArtiREST reaches as high as 220 in Fig. 6(b), much higher than the previous experiments in Fig. 5 (about 40 there). This is because in this experiment, we just consider the number of CPU in a single machine but let the RAM free (32 GB) when performing the vertical scalability evaluation.

5.4 Dynamic Scalability

In the previous discussions, we start a certain number of process engine instances simultaneously and test the maximum TPS as its capability. In the real world, one interesting problem is the ability to dynamically adjust the engine instances according to the current system loads. Dynamically scaling is attractive because it adapts to the growing of loads without restarting the system. Figure 7 shows the result of a simple dynamic scalability experiment. It depicts the behaviors of TPS and ART against the number of users as process engine instances steadily increased. From the three subgraphs, we can see that when the number of users reaches 173, 390, and 510 respectively, the system begins to saturate. However, new engines start and join the cluster so that it breaks all saturation point by lowering response time and increasing TPS. Of course, the result is not perfect yet. First, the latency of starting an engine is too high (more than 10 s). Second, we dont know the perfect timing to start new engines. However, it does provide some findings that the ArtiREST is dynamically scalable. This indicates that the framework is much promising in pursuit of better scalability. Scaling dynamically is difficult, which involves loads monitoring, loads prediction, and scaling without interrupting and latency. We will study these issues in future work.

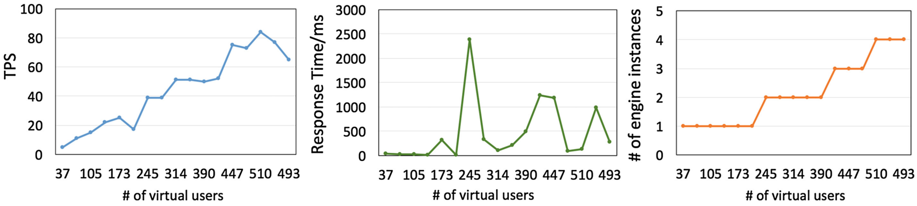


Fig. 7. Dynamically scaling the process engine instances according to the loads

6 Related Work

The study in this paper is about the scalability of artifact-centric BPM frameworks. We could classify related work into three categories, i.e. artifact-centric BPM implementation frameworks, BPM based on RESTful style, and a combination of artifact-centric approach that following RESTful style.

Since the birth of the artifact-centric approach [13], a number of implementation frameworks have emerged. Barcelona [7] and ACP framework [12] are arguably the most important frameworks among them. Barcelona [7] is a homegrown environment from IBM research, the inventor of the artifact-centric approach. The formal model GSM [8] and its open-sourced implementation BizArtifact[2] could be considered as the benchmark in the area of artifact-centric BPM. In the open-source world, there is another implementation ACP [12]. Besides its public accessibility, we prefer ACP because it reflects the original idea [13] of artifact-centric approach faithfully. Hence, in this study, we adopt the ACP model and adapt it to RESTful style. When evaluating the scalability, we choose them as our references. As indicated in Fig. 6(b) we found that both of them are underperforming in vertical scalability, aside from inability

[2] BizArtifact: https://sourceforge.net/projects/bizartifact.

in horizontal scalability. Similarly, our previous frameworks, ArtiFlow [10] and EZ-Flow [3] have no concerns about scalability.

Parallel with artifact-centric BPM researches, there exist some explorations of application of REST architectural styles on activity-centric BPM [16,21]. Besides the perspective difference, most works in this direction focus on design principles or guidelines, instead of the concrete framework implementation.

The work of Kumaran *et al.* [9] is closest to the our work. They propose a RESTful architecture for service-oriented business process execution in which business entities are modeled as Mealy machines, and the framework uses resource brokers to handle service requests and manage business entities. Contrasting to their indirect approach subject to the REST constraints, ArtiREST gives a straight answer. Moreover, they focus on the discussion of desirable properties, such as flexibility, interoperability, and scalability, but leave the implementation aside. As result, they provide neither experiments nor evidence to prove their claims in scalability.

7 Conclusion

In this paper, we studied the scalability issue in artifact-centric BPM systems, which was neglected by other researches on artifact-centric approaches. We then proposed a distributed framework ArtiREST based on the REST architectural style. Extensive experiments demonstrate the advantage of ArtiREST in both horizontal and vertical scalabilities over mainstream implementations of artifact-centric BPM. It can serve as a convincing evidence that the RESTful framework is effective and promising to reach a scalable artifact-centric BPM system.

In future work, we are going to improve the framework to support more management aspects of BPM such as performance monitoring and transaction support. It is also interesting to study how to scale out/in and up/down dynamically of the number of process engine instances according to current loads and loads prediction.

Acknowledgments. The work is supported by National NSFC (No. 60873115), Shanghai Science and Technology Development Funds (No. 13dz2260200 & No. 13511504300), and National Hi-Tech. Project (2012AA02A602).

References

1. Bhattacharya, K., Gerede, C.E., Hull, R., Liu, R., Su, J.: Towards formal analysis of artifact-centric business process models. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 288–304. Springer, Heidelberg (2007). doi:10.1007/978-3-540-75183-0_21
2. Bondi, A.B.: Characteristics of scalability and their impact on performance. In: Proceedings of the 2nd International Workshop on Software and Performance, pp. 195–203. ACM (2000)
3. Chen, Y., Xu, W., Zhang, L., Su, J.: EZMS: a workflow management system for EZ-Flow (in Chinese). Comput. Technol. Dev. **22**(12), 1–6 (2012)

4. Cohn, D., Hull, R.: Business artifacts: a data-centric approach to modeling business operations and processes. Bull. IEEE Comput. Soc. Tech. Committee Data Eng. **32**(3), 3–9 (2009)
5. Fielding, R.T., Taylor, R.N.: Principled design of the modern web architecture. ACM Trans. Internet Technol. (TOIT) **2**(2), 115–150 (2002)
6. Fielding, R.T.: Architectural styles and the design of network-based software architectures. Ph.D. thesis, University of California, Irvine (2000)
7. Heath III, F.T., Boaz, D., Gupta, M., Vaculín, R., Sun, Y., Hull, R., Limonad, L.: Barcelona: a design and runtime environment for declarative artifact-centric BPM. In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (eds.) ICSOC 2013. LNCS, vol. 8274, pp. 705–709. Springer, Heidelberg (2013). doi:10.1007/978-3-642-45005-1_65
8. Hull, R., Damaggio, E., De Masellis, R., Fournier, F., Gupta, M., Heath III., F.T., Hobson, S., Linehan, M., Maradugu, S., Nigam, A., et al.: Business artifacts with guard-stage-milestone lifecycles: managing artifact interactions with conditions and events. In: Proceedings of the 5th ACM International Conference on Distributed Event-Based System, pp. 51–62. ACM (2011)
9. Kumaran, S., Liu, R., Dhoolia, P., Heath, T., Nandi, P., Pinel, F.: A RESTful architecture for service-oriented business process execution. In: Proceedings of IEEE International Conference on e-Business Engineering, pp. 197–204 (2008)
10. Liu, G., Liu, X., Qin, H., Su, J., Yan, Z., Zhang, L.: Automated realization of business workflow specification. In: Dan, A., Gittler, F., Toumani, F. (eds.) ICSOC/ServiceWave 2009. LNCS, vol. 6275, pp. 96–108. Springer, Heidelberg (2010). doi:10.1007/978-3-642-16132-2_9
11. Menascé, D.A.: Load testing of web sites. IEEE Internet Comput. **6**(4), 70–74 (2002)
12. Ngamakeur, K., Yongchareon, S., Liu, C.: A framework for realizing artifact-centric business processes in service-oriented architecture. In: Lee, S., Peng, Z., Zhou, X., Moon, Y.-S., Unland, R., Yoo, J. (eds.) DASFAA 2012. LNCS, vol. 7238, pp. 63–78. Springer, Heidelberg (2012). doi:10.1007/978-3-642-29038-1_7
13. Nigam, A., Caswell, N.S.: Business artifacts: an approach to operational specification. IBM Syst. J. **42**(3), 428–445 (2003)
14. Pautasso, C.: BPMN for REST. In: Dijkman, R., Hofstetter, J., Koehler, J. (eds.) BPMN 2011. LNBIP, vol. 95, pp. 74–87. Springer, Heidelberg (2011). doi:10.1007/978-3-642-25160-3_6
15. Pautasso, C.: RESTful web services: principles, patterns, emerging technologies. In: Bouguettaya, A., Sheng, Q.Z., Daniel, F. (eds.) Web Services Foundations, pp. 31–51. Springer, New York (2014)
16. Pautasso, C., Wilde, E.: Push-enabling RESTful business processes. In: Kappel, G., Maamar, Z., Motahari-Nezhad, H.R. (eds.) ICSOC 2011. LNCS, vol. 7084, pp. 32–46. Springer, Heidelberg (2011). doi:10.1007/978-3-642-25535-9_3
17. Tan, W., Hsu, J., Pinn, F.: Method and system for token-based authentication (Feb 23 2001), US Patent App. 09/792,785
18. Webber, J., Parastatidis, S., Robinson, I.: REST in Practice: Hypermedia and Systems Architecture. O'Reilly Media, Inc., Sebastopol (2010)
19. Weske, M.: Business Process Management. Concepts, Languages, Architectures. Springer, Heidelberg (2012)
20. van der Aalst, W.M.P.: Business process management: a comprehensive survey. ISRN Softw. Eng. **2013**, 1–37 (2013). doi:10.1155/2013/507984
21. Xu, X., Zhu, L., Kannengiesser, U., Liu, Y.: An architectural style for process-intensive web information systems. In: Chen, L., Triantafillou, P., Suel, T. (eds.) WISE 2010. LNCS, vol. 6488, pp. 534–547. Springer, Heidelberg (2010)

Bridging Semantic Gap Between App Names: Collective Matrix Factorization for Similar Mobile App Recommendation

Ning Bu(✉), Shuzi Niu, Lei Yu, Wenjing Ma, and Guoping Long

Institute of Software, Chinese Academy of Sciences,
4# South Fourth Street, Zhong Guan Cun, Beijing, China
{buning13,shuzi,yulei,wenjing,guoping}@iscas.ac.cn

Abstract. With the increase of mobile apps, i.e. applications, it is more and more difficult for users to discover their desired apps. Similar app recommendation, which plays a critical role in the app discovering process, is of our main concern in this paper. Intuitively, name is an important feature to distinguish apps. So app names are often used to learn the app similarity. However, existing studies do not perform well because names are usually very short. In this paper, we explore the phenomenon of the ill performance, and dive into the underlying reason, which motivates us to leverage additional corpus to bridge the gap between similar words. Specifically, we learn app representation from names and other related corpus, and formalize it as a collective matrix factorization problem. Moreover, we propose to utilize alternating direction method of multipliers to solve this collective matrix factorization problem. Experimental results on real-world data sets indicate that our proposed approach outperforms state-of-the-art methods on similar app recommendation.

Keywords: Mobile application · Similar app recommendation · Collective matrix factorization · Alternating direction method of multipliers

1 Introduction

Mobile apps, i.e. applications, are ubiquitous along with the smart phones, and play a more and more important role in our daily life. However, it is a daunting task to discover the desired apps for users from the vast number of apps. Similar app recommendation is such a useful way to enhance user experience. For instance, a list of similar apps is presented for each app in Google Play, users can obtain other similar or related apps with minimal effort when browsing a certain app. Thus we focus on how to obtain such a list for similar app recommendation.

Two apps are supposed to be similar if their meta information is semantically related. This is called High Level Software Similarity in previous work [3]. Here app meta information means the name, description, rating, reviews, screen shots and all other information describing the app itself. Among all kinds of app meta information, name is often used as the meaningful identifier to distinguish

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 324–339, 2016.
DOI: 10.1007/978-3-319-48743-4_26

different apps. App developers use name to summarize one app as two or three discriminative, impressive words. App users use names to search a particular app. According to a survey [27], 88 % query terms are from app names. Generally, name functions as keywords of all the meta information for each app.

As important feature, name arouses much research interest, but few of them make full use of names in app representation learning. Because the app name usually contains a small number of words, two or three for example, the word independence assumption in traditional vector space model is supposed to be false for short texts. Existing studies on the similarity between short texts usually leverage additional information to help model the word dependency, but such additional information is often obtained with great effort, especially for those private information such as query log. Besides, these methods usually relies on the particular kind of information. Thus how to learn app representation from names systematically is a challenge in the similar app recommendation task.

We first explore the reason of introducing additional corpus to model word dependency. Then, we propose a systemic method to utilize this additional corpus to help learn representation from names. The word dependency is modeled as the word co-occurrence in the additional corpus. We jointly utilize the vector space model and word dependency to derive the app representation by formalizing it as a collective matrix factorization problem [20]. In this paper, we take description for example because it can be as easily obtained from app stores as app names. Finally, we propose a new optimization algorithm to solve this collective matrix factorization problem. In contrast to the optimization algorithm proposed in [20], the alternating direction method of multiplier avoids the computation of Hessian matrix and converges faster without loss of performance. We investigate our proposed approach on real-world data sets crawled from Google Play. Experimental results show that our proposed approach outperforms state-of-the-art methods in the similar app recommendation task.

Our method will help app developers promote their new apps, help app stores attract more app users, and help app users find their desired apps. In all, the major contributions of this paper lie in the following three aspects:

- the exploration of the necessity of introducing additional corpus to help representation learning from names.
- the formalization of the app representation learning with the help of additional corpus as the collective matrix factorization problem.
- the proposal to adopt alternating direction method of multiplier to solve the problem above.

The rest of the paper is organized as follows: Sect. 2 discuss related work; Sect. 3 describe our motivation for introducing additional corpus for names; Sects. 4 and 5 present the problem formulation and the optimization algorithm respectively; Sect. 6 show the empirical results; Sect. 7 concludes the paper.

2 Related Work

In this paper, we investigate the role that app name plays in the app representation for similar app recommendation task. As a typical kind of short text, app name can be treated as document, sentence, words or phrase, but not exactly the same. Statistical patterns of human word usage can be used to figure our what people mean. In other words, if units of text have similar vectors in a text frequency matrix, then they tend to have similar meanings. This is statistical semantics hypothesis [15]. According to different kinds of statistical semantics hypothesis, we review the related work in terms of document similarity based on bag of words hypothesis and word similarity based on distributional hypothesis.

Document Similarity. Treating the document similarity as its content similarity, there are a bundle of classical methods to model the document content, like TF-IDF [17], LSI [19], LDA [2]. LSI [19] employs matrix factorization methods of the document-term matrix, such as singular value decomposition (SVD), to generate the low-dimensional document representations. LDA [2], short for Latent Dirichlet Allocation, is a three-level hierarchical Bayesian model of text corpora, in which each document in the collection is modeled as a finite mixture over an underlying topics, and each topic is modeled as an infinite mixture over words of topic probabilities. All the three methods are mainly based on vector space model [6], which is supposed to be the first practical, useful algorithm for exacting semantic information from word usage [15]. Bag of words hypothesis says that If documents have similar column vectors in a document-term matrix, then they tend to have similar meanings. To enrich the short texts, various related studies [10,12,18,23] propose to leverage external resources, such as web search results, to improve the semantical similarity between text segments.

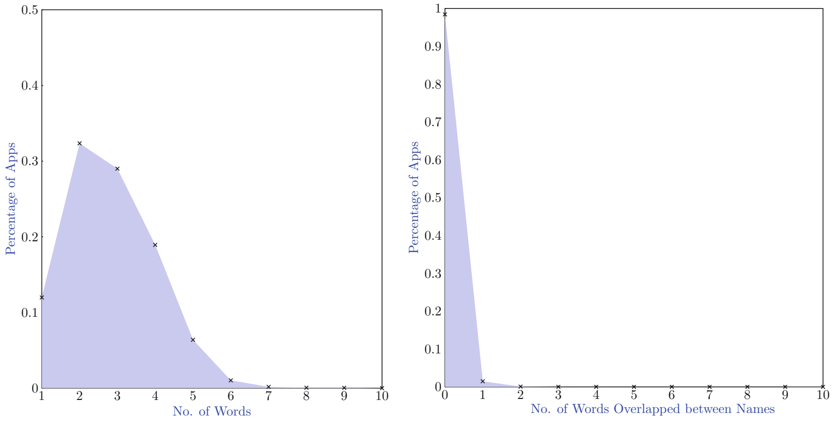
Phrase/Word Similarity. Words that occur in similar contexts tend to have similar meanings [7,19]. Distributional hypothesis states that if words have similar row vectors in a word-context matrix, then they tend to have similar meanings. A lot of recent studies are based on this hypothesis, such as [9,16,21]. word2vec [21] learns word vectors from word-context matrix through neural network model, then it is extended to sentence and paragraph [16], namely Doc2vec in this paper. The focus shift from document to word provides more possibilities, for instance the context may be words, phrases, sentences, paragraphs, chapters, documents, sequences of characters and so on.

App Similarity. Existing studies [3,4,14,25] focus on the app representation learning from meta information in different tasks. For **app classification**, Zhu et al. [25] enrich the app representation by exploiting the additional Web knowledge from the Web search engine. For **app search**, AppLDA [14] is proposed to introduce user reviews to bridge the vocabulary gap between app developers and users to learn app representation by extending LDA. MobileWalla [1], a mobile application search engine, computes the semantic similarity between app based on the WordNet [5]. Panorama [4], a semantic-aware application search framework, propose the App Topic Model that integrates the text, link and category information in order to discover the latent semantics from apps. For **similar app recommendation**, SimApp [3] is the state-of-the-art framework to employ multiple heterogeneous kinds of app information to detect similar apps.

Different from the related studies on document and phrase or word similarity, our method takes the word dependency into consideration for the short app name the and leverages external source without too much effort. Distinguished from existing studies on app similarity, we put emphasis on the importance of names in learning app representation. SimApp [3] considers the app name as a short string of characters and utilizes the well-known string kernel [8], referred to as SSK (subsequence kernel) to model the similarity between apps as string matching without considering their semantics.

3 Motivation

App name is such an important feature that cannot be ignored in the app presentation learning. First, app names are usually originated from the summarization of the description, so words in app names are representative and less noisy. Second, though not unique, app names are often used as the identifier where one app is different from the others, so words in app names are discriminative. Finally, users identify similar apps usually by names intuitively. However, experimental results in existing studies [3,4,14,25] show app names do not play the most significant role in similar app recommendation. This phenomenon arouses our curiosity to study this problem.



(a) Word No. Distribution (b) Overlapped Word No. Distribution

Fig. 1. Word statistics in crawled app names

Words in app names are usually representative. Here we mean words in app names can be treated as the keywords of app description texts. We simply use the TF-IDF to predict the keywords in app description texts, and top k words are selected. Therefore the accuracy of this simple keyword extraction method can be coarsely estimated as the representativeness of words in app names, such as Precision@1 and Precision@2. Statistical results on our crawled more than 20 thousand apps show that Precision@1 and Precision@2 are 5 %, 4 % respectively.

Words in app names are often discriminative. Simply we use the literal difference between two app names to measure the discrimination of words, the word overlap between any two app names. On average, the number of words overlapped between any two app names is nearly 0.02. Among 200 million app pairs, 98 % pairs with no word in common. These results suggest app developers tend to use different words from existing app names even when two apps are semantically related. This characteristic of word usage will lead to the failure of word independence assumption in measuring the similarity between app names.

However, the number of words in app names are always smaller than ever. For instance, a common short text is microblog with the averaged number of words 5 or so. Statistical results on our crawled apps show that the averaged number of words is 2.8 for each app name in Fig. 1. In Fig. 1, more than 90 % apps have no more than 4 words in their names, and almost all the apps have no more than 6 words in their names. The small number of words in app names will cause the sparsity of models based on word independence assumption directly.

Generally, both the discriminative words and the smaller number of words in app names make it more difficult to learn a better semantic space for apps from names. Therefore we cannot connect "Uber" and "Yandex.Taxi" by looking at words alone. With the help of auxiliary text documents where these words co-occur frequently, we may establish a strong semantic similarity between words. As a result, it is necessary to incorporate auxiliary word dependency in learning app representation from names.

4 Methods

We first give a formal definition of app representation learning problem. Then we show this problem can be reduced to the Collective Matrix Factorization problem, denoted as CMF. Finally, app vectors obtained from CMF are used to compute the similarity between apps for similar app recommendation task.

4.1 Problem Definition

For each app $a \in \mathcal{A}$, there is a collection of information describing a, such as name E_a, description D_a. $\mathcal{A}$ is a set of n mobile apps. In this paper, we focus on the problem as follows.

Definition 1 (App Representation Learning). *Let E describe app names and L describe the semantic relation between two words for all apps in $\mathcal{A}$, our app representation learning problem aims at learning a latent semantic space for each app from both E and L.*

Given the name-word matrix E for all the apps in $\mathcal{A}$ and their associated auxiliary document-word matrix L, we hope to bridge the gap between word that are potentially semantically related. Here we use the app description as the auxiliary documents for example. As illustrated in Fig. 2, we construct a two-layer bipartite graph among app names, words and app descriptions.

Specifically, the left layer of bipartite graph is used to represent the relationship between names and words. Each app name can be represented as a vector of word occurrences and some app names share one or multiple word. If names of two apps have one or more words in common, they are supposed to be semantically related. Similarly, if two words co-occur in the same auxiliary document, they will be semantically related. This relationship between words and auxiliary documents is represented in the right layer of bipartite graph.

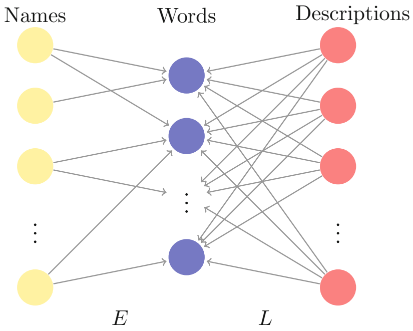


Fig. 2. Two-layer bipartite graph: name-word bipartite graph E and description-word bipartite graph L

4.2 Collective Matrix Factorization

We use the typical vector space model to obtain the name-word matrix $E \in \mathbb{R}^{n\times l}$, where n is the number of apps, and l is the size of app name dictionary. $E = (e_{ij})_{n\times l}$ where e_{ij} means TFIDF of word j in the name of app i, $E_{i\cdot}$ is the representation of app i in the original space. We derive the latent semantic space $U \in \mathbb{R}^{n\times k}$ by LSI [19] as $E = UV_1^T$, where $U_{i\cdot}$ is the latent semantic representation of app i, and k is the dimension of this latent semantic space.

$L \in \mathbb{R}^{n\times l}$ is a document-word matrix, where $L = (l_{ij})_{n\times l}$, l_{ij} means the count of word j in the description of app i, such as TF-IDF, TF and binary value. Here we use the TF value for example. $L_{i\cdot}$ is the representation of the description of app i. Using Latent Semantic Indexing [19], we obtain the latent semantic representation by $L = WV_2^T$.

In order to bridge the gap between words that are likely to be semantically related, we propose to learn the latent semantic representation U by decomposing E and L jointly with the constraint $V_1 = V_2$. This is called collective matrix factorization, denoted as CMF, which was proposed by Singh and Gordon [20].

Our **App** representation **Le**arning problem can be reduce the following CMF problem in Eq. (1), denoted as AppLe-CMF[1].

$$\begin{aligned} \min \quad & \lambda_E \|E - UV_1^T\|_F^2 + \lambda_L \|L - WV_2^T\|_F^2 + \Omega(U, V_1, V_2, W) \\ s.t. \quad & V_1 = V_2 \end{aligned} \tag{1}$$

In the optimization objective function of AppLe-CMF as Eq. (1), λ_E and $\lambda_L (0 \le \lambda_E, \lambda_L \le 1, \lambda_E + \lambda_L = 1)$ are trade-off parameters between the factorization error of E and L. $\|M\|_F$ is the Frobenius Norm of M. $\Omega(U, V_1, V_2, W)$ is the regularized term to control the complexity of U, V_1, V_2 and W. In this paper, we defined this term as $\gamma_1 \|U\|_F^2 + \gamma_2 \|V_1\|_F^2 + \gamma_3 \|V_2\|_F^2 + \gamma_4 \|W\|_F^2$.

4.3 App Similarity

We obtain the latent semantic representation U for all apps by optimizing the objective function of AppLe-CMF. To apply it to the similar app recommendation, we employ the cosine similarity to compute the semantically similarity $s(a_i, a_j)$ between any two apps as in Eq. (2). For each query app, top d apps with higher similarity are recommended.

$$s(a_i, a_j) = \frac{U_{i\cdot} U_{j\cdot}}{\|U_{i\cdot}\| \|U_{j\cdot}\|} \tag{2}$$

5 Optimization Algorithm

The remaining question is how to optimize the objective function of AppLe-CMF in Eq. (1). First, we introduce the augmented Lagrangian function for better convergence. Second, we apply the alternating direction method of multiplier to solve the collective matrix factorization problem. Finally, we analyze the time complexity of our optimization method.

5.1 Augmented Lagrangian Function

We transform the constrained optimization problem AppLe-CMF into the unconstrained optimization problem by introducing the Lagrangian function

$$\lambda_E \|E - UV_1^T\|_F^2 + \lambda_L \|L - WV_2^T\|_F^2 + \Omega(U, V_1, V_2, W) + \Lambda \bullet (V_1 - V_2),$$

where Λ is the Lagrangian multiplier. The scalar product $\bullet$ is the sum of all element-wise products of two matrices A and B of the same size, i.e., $A \bullet B = \sum_{i,j} a_{ij} b_{ij}$. Although this Lagrangian function is non-convex with all the four matrices U, V_1, V_2 and W, it is convex with respect to any one matrix while fixing the other three.

Existing alternating methods [22] optimize one matrix by fixing the others iteratively until the results converges. The only difference among different

[1] Core code is available at https://github.com/bnn2010/iscas2016_AppSimilarity.

alternating methods lies in the update step. Some methods utilize the conjugate gradient descent step [26], other methods employ the Newton-Raphson step [20].

Compared with the classical alternating methods [20,22,26], the ADMM algorithm converge much faster due to the addition of the quadratic term $\|V_1 - V_2\|_F^2$, which can be interpreted as quadratic Tikhonov regularization [13]. This damping term encourages $V_1^{(t)}$ not to be very far from $V_2^{(t)}$ after t iterations. As the ADMM algorithm converges, $V_1^{(t)}$ gets close to $V_2^{(t)}$, so the effect of the quadratic regularization goes to zero. In other words, the quadratic regularization contributes a term to the gradient that decreases to zero as the algorithm proceeds. Therefore, the augmented Lagrangian function of the original problem AppLe-CMF in Eq. (1) is

$$\begin{aligned}\mathcal{L}(U, V_1, V_2, W, \Lambda) = &\lambda_E \|E - UV_1^T\|_F^2 + \lambda_L \|L - WV_2^T\|_F^2 + \Omega(U, V_1, V_2, W)\\ &+ \Lambda \bullet (V_1 - V_2) + \alpha \|V_1 - V_2\|_F^2,\end{aligned} \tag{3}$$

where Λ is a lagrangian multiplier and $\alpha > 0$ is a penalty parameter. The Lagrangian multiplier Λ is a dual variable associated with the consensus constraint $V_1 = V_2$, which means the violation degree of the constraint.

5.2 Alternating Direction Method of Multipliers for Collective Matrix Factorization

The alternating direction method of multipliers for the augmented Lagrangian function in Eq. (3) is derived by successively minimizing $\mathcal{L}$ with respect to U, V_1, V_2, W, one at a time while fixing others at their most recent values, then updating the multiplier Λ. For each iteration $t+1$, the update includes two parts:

1. update U, V_1, V_2 and W

$$\begin{aligned}U^{(t+1)} &= \arg\min \mathcal{L}(U, V_1^{(t)}, V_2^{(t)}, W^{(t)}, \Lambda^{(t)})\\ V_1^{(t+1)} &= \arg\min \mathcal{L}(U^{(t+1)}, V_1, V_2^{(t)}, W^{(t)}, \Lambda^{(t)})\\ V_2^{(t+1)} &= \arg\min \mathcal{L}(U^{(t+1)}, V_1^{(t+1)}, V_2, W^{(t)}, \Lambda^{(t)})\\ W^{(t+1)} &= \arg\min \mathcal{L}(U^{(t+1)}, V_1^{(t+1)}, V_2^{(t+1)}, W, \Lambda^{(t)})\end{aligned} \tag{4}$$

2. update the dual variable Λ

$$\Lambda^{(t+1)} = \Lambda^{(t)} + 2\eta\alpha(V_1^{(t+1)} - V_2^{(t+1)}) \tag{5}$$

where η is a step length.

Considering the convexity of each primal variable to $\mathcal{L}$, the minimization in each step is achieved by the local minimum using the most recent values of the other primal variables and the dual variables. Thus we obtain the closed form of the minimization problem in each step as Eq. (6).

$$\begin{aligned}
U^{(t+1)} &= \arg\min\{\lambda_E\|E - UV_1^{(t)T}\|_F^2 + \gamma_1\|U\|_F^2\} \\
&= \lambda_E E V_1^{(t)}(\lambda_E V_1^{(t)T}V_1^{(t)} + \gamma_1 I_{k\times k})^{-1} \\
V_1^{(t+1)} &= \arg\min\{\lambda_E\|E - U^{(t+1)}V_1^T\|_F^2 + \gamma_2\|V_1\|_F^2 + \varLambda^{(t)} \bullet V_1 + \alpha\|V_1 - V_2^{(t)}\|_F^2\} \\
&= (2\lambda_E E^T U^{(t+1)} + 2\alpha V_2^{(t)} - \varLambda^{(t)})(2\lambda_E U^{(t+1)T}U^{(t+1)} + (2\gamma_2 + 2\alpha)I_{k\times k})^{-1} \\
V_2^{(t+1)} &= \arg\min\{\lambda_L\|L - W^{(t)}V_2^T\|_F^2 + \gamma_3\|V_2\|_F^2 - \varLambda^{(t)} \bullet V_2 + \alpha\|V_1^{(t+1)} - V_2\|_F^2\} \\
&= (2\lambda_L L^T W^{(t)} + 2\alpha V_1^{(t+1)} + \varLambda^{(t)})(2\lambda_L W^{(t+1)T}W^{(t+1)} + (2\gamma_3 + 2\alpha)I_{k\times k})^{-1} \\
W^{(t+1)} &= \arg\min\{\lambda_L\|L - WV_2^{(t+1)T}\|_F^2 + \gamma_1\|W\|_F^2\} \\
&= \lambda_L L V_2^{(t+1)}(\lambda_L V_2^{(t+1)T}V_2^{(t+1)} + \gamma_4 I_{k\times k})^{-1}
\end{aligned} \tag{6}$$

Algorithm 1. Alternating Direction Method of Multipliers for Collective Matrix Factorization

```
Input:
   the matrix $E, L \in \mathbb{R}^{n\times l}$;
Output: the latent semantic space $U \in \mathbb{R}^{n\times k}$.
begin
set $V_1$,$V_2$ as a random matrix
set $\alpha$,$\eta > 0$,
set $U$, $W$ and $\varLambda$ as zero matrix of appropriate sizes
while $\Delta\mathcal{L}(U, V_1, V_2, W, \varLambda) \geq \epsilon$ do
     update $(U, V_1, V_2, W)$ as Eq. (6);
     update $\varLambda$ as Eq. (5);
end while
return $U$
end
```

As shown in Algorithm 1, the stopping criterion is met when the following condition is satisfied: $\frac{|\mathcal{L}_{t+1} - \mathcal{L}_t|}{\mathcal{L}_0} \leq \epsilon$. $\mathcal{L}_{t+1}$ and $\mathcal{L}_t$ are objective values in the iteration $t+1$ and t respectively. ϵ is usually a small value.

5.3 Time Complexity

Each iteration in the Algorithm 1 consists of updating the five matrices U, V_1, V_2, W and $\varLambda$. Then we analyze the time complexity for each iteration.

The most time-consuming computation in updating U lies in the multiple matrices multiplication and the matrix inversion in Eq. (6). Time complexity for obtaining the matrix to be inverse and computing the inversion of a matrix with size $k \times k$ is about $\mathcal{O}(lk^2)$ and $\mathcal{O}(k^3)$ respectively. Computing the multiplication of three matrices with size $n \times l$, $l \times k$ and $k \times k$ separately cost about $\mathcal{O}(nlk + \max\{n, l\}k^2)$. Therefore the overall complexity is $\mathcal{O}(k^3 + lk^2 + \max\{n, l\}k^2 + nlk)$.

Similarly, the time complexity for updating V_1, V_2, W and $\varLambda$ is $\mathcal{O}(k^3 + lk^2 + nk^2 + nlk)$, $\mathcal{O}(k^3 + lk^2 + nk^2 + nlk)$, $\mathcal{O}(k^3 + lk^2 + \max\{n, l\}k^2 + nlk)$

and $\mathcal{O}(lk)$ separately. For each iteration in Algorithm 1, the time complexity is $\mathcal{O}(k^3+lk^2+\max\{n,l\}k^2+nlk)$. Its leading complexity $\mathcal{O}(nlk)$ is the same as that in gradient descent method. In this sense, the ADMM is nearly as efficient as the gradient descent method. Suppose the algorithm converges after T iterations, the overall complexity is $\mathcal{O}(nlkT)$, which is comparable to those of baseline methods, such as LDA [2], LSI [19] and Doc2vec [16].

6 Experiments

First we present our crawled apps from Google Play, which are used as our datasets, and other experimental settings. Then we conduct comprehensive experiments on our datasets in terms of accuracy analysis and parameter sensitivity analysis in similar app recommendation task. To make a further step towards the performance improvement, we analyze an illustrative example in detail and compare the word vector for better understanding.

6.1 Experimental Setting

Datasets. We establish our original data set in two steps. First, we use the app meta-information provided in [3] and obtain 15,282 apps which belong to 42 categories, after removing some noisy apps. For each app, name and description are preprocessed by removing the punctuation, stop words and low-frequency words, stemming. Thus there are 3,835 distinct words in names and 41,773 words in descriptions. Then, we crawl similar app lists presented in Google Play Store because the test collection is not provided in [3]. The ground truth for app similarity is generated from our crawled tens of thousands of similar app lists. Like [3], two apps are considered to be similar if they co-occur in the same similar app list more than once.

Baselines. In this section, we explain the process of our empirical study and briefly review several state-of-art algorithms we compared to as baselines. We compared our proposed AppLe-CMF with three kinds of typical baseline methods: (1) Document Similarity Models: TF-IDF [17], LSI [19], LDA [2]; (2) Word Similarity Models: Doc2vec [16]; (3) String Similarity Models: the string kernel [8] used in SimApp [3] denoted as SimApp_S. For those latent space models like LSI, LDA, Doc2vec and AppLe-CMF, we choose the dimension of latent space from 10 to 100 with step 10 and from 200 to 1000 with step 100. Additionally, our proposed method AppLe-CMF need to adjust a trade-off parameter from 0.1 to 0.9 with step 0.1. All these performances shown in the following sections are in the best configuration. We set α as $1.91\times10^{-4}\frac{\max\{\|E\|_F,\|L\|_F\}\max\{n,l\}}{k}$ like [24]. ϵ in the stopping criterion is 10^{-6} and η is 1.618 as [24] in our experiments.

Evaluation. For rank-based evaluation, we set each app as query app, and its similar apps can be obtained from the similar app lists. After removing those query apps with the number of similar apps less than 5, we obtain $4,457$

query apps. Based on the unreliable assumption that apps out of app a's similar app list are dissimilar to a, we choose dissimilar apps for each query app as in SimApp [3]. To avoid the bias of the unreliable assumption, we randomly select different numbers of dissimilar apps, such as 50, 100, 200. Therefore we obtain three test data sets with each query app corresponding to a list of more than 5 similar apps (labeled with 1) and a number of dissimilar apps (labeled with 0), denoted as AppSet50, AppSet100, AppSet200. We evaluate the performance of various methods for similar app recommendation with rank-based measures [11], i.e. Precision@k(denoted as P@k) and NDCG@k.

6.2 Ranking Accuracy Analysis

With P@5 and NDCG@5 as evaluation measures, the best averaged performance over all the query apps is shown in Table 1. For all the three data sets, the optimal parameter setting of k is 900, 200, 300, 1000 for LSI, LDA, Doc2vec and AppLe-CMF separately and the optimal λ_L is 0.9 for AppLe-CMF. Statistical significance tests are also done for performance comparisons.

Table 1. Performance comparison between AppLe-CMF and baseline methods

Approach	AppSet50		AppSet100		AppSet200	
	P@5	NDCG@5	P@5	NDCG@5	P@5	NDCG@5
Doc2vec	0.232847	0.232050	0.140273	0.141457	0.079649	0.079178
TFIDF	0.669598	0.702618	0.624680	0.658857	0.583980	0.616297
LSI	0.698093	0.726622	0.653130	0.680771	0.607494	0.633038
LDA	0.641239	0.666057	0.576576	0.598272	0.496433	0.513752
SimApp_S	0.702580	0.733794	0.652188	0.683939	0.597846	0.628773
AppLe-CMF	0.714158[a]	0.738584[a]	0.669778[a]	0.692549[a]	0.613058[a]	0.634468[a]

[a] Performance differences between AppLe-CMF and any baseline are statistically significant with p-value < 0.01 for paired t-tests.

As shown in Table 1, our proposed method AppLe-CMF achieves the best performance on all the three data sets among all the methods. For example, P@5 of AppLe-CMF on AppSet100 is 6.7 %, 2.3 %, 11.4 %, 1.6 %, 377.48 % higher than TF-IDF, LSI, LDA, SimApp_S and Doc2vec separately. Obviously, both typical document similarity models (i.e., TF-IDF, LSI, LDA) and word similarity model (i.e. Doc2vec) perform not so well as string similarity model (i.e. SimApp_S) mainly because the string kernel used in SimApp makes full use of the limited words of names in terms of characters. Generally, the introduction of word dependency brings about the performance improvement of AppLe-CMF.

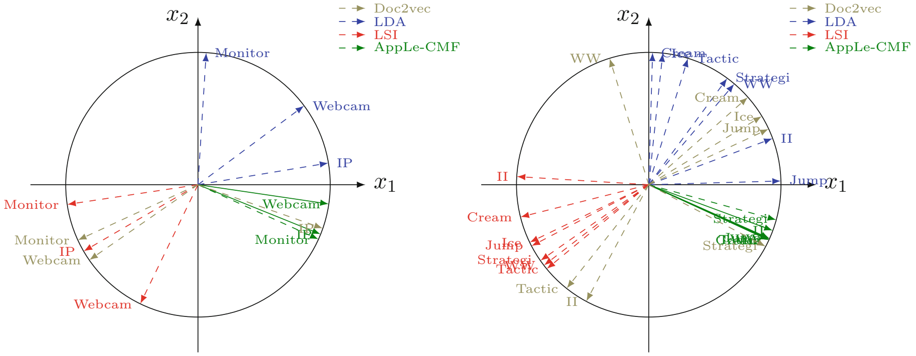
With the increase of the dissimilar app number from each dataset in Table 1, the performance of each method decreases. Taking SimApp_S for example, NDCG@5 on AppSet100 and AppSet200 is 6.8 % and 14.3 % less than that on

AppSet50 respectively. This performance deduction can be explained by two reasons. One is that the similar app recommendation task becomes more difficult as the dataset size increases. The other is the unreliable assumption that there is some similar apps labeled with 0 as mentioned before. In that case, these noisy labels have no effect on our performance comparison results.

Table 2. Similar apps for "Uber" from AppLe-CMF and baselines

	LDA	LSI	AppLe-CMF	SimApp_S	TFIDF	Doc2vec
1	Duel Quiz	CNHandwriting for GO Keyboard	Taxibeat Free taxi app	Battery Widget Lightsaber	Alaska Airlines Travel App	GalaxSim Unlock
2	SugarSync	GalaxSim Unlock	inTaxi: order taxi in Russia	Het Weer in Nederland	The Economic Times	Easy Taxi, Cab App for Drivers
3	Lyft	inTaxi: order taxi in Russia	Yandex.Taxi	Water Drop Live Wallpaper	Het Weer in Nederland	Hill Climb Racing
4	BiTaksi	Pinball Pro	Tappsi Taxista	Meru Cabs	Essential Memo	TapTaxi
5	PipeRoll	Duel Quiz	EST: Call Taxi?	Calorie Counter	Yandex.Taxi	BusyBox X

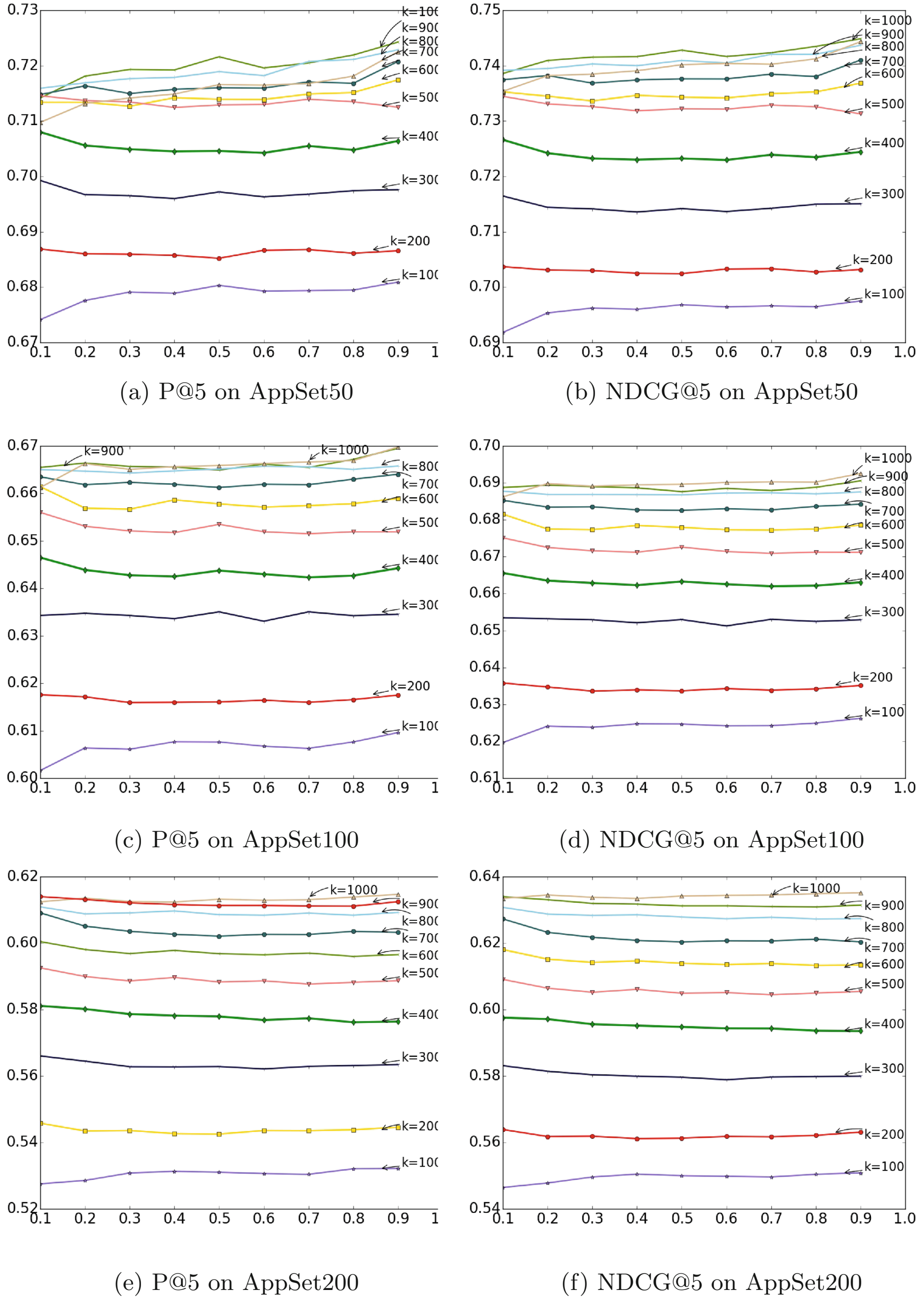
Specifically for the query app "Uber", which is a well-known taxi booking app, we list top 5 apps names for each method in Table 2. One app is similar (not similar) to the query app if its name is in black (red) color. Note that in Table 2, "Taxibeat Free taxi app", "BiTaksi", "Yandex.Taxi", "Lyft" etc. are all taxi booking apps. This qualitative example illustrates the better performance of AppLe-CMF. TF-IDF and SimApp_S cannot capture the latent semantic relation between "Uber" and "taxi", which are well related in AppLe-CMF. The semantic similarity between word vectors may explain the better performance of AppLe-CMF, which inspires us to explore the underlying reason further.



(a) tinyCam Monitor

(b) Strategy & Tactics: WW II Free

Fig. 3. Word vectors in latent space



(a) P@5 on AppSet50

(b) NDCG@5 on AppSet50

(c) P@5 on AppSet100

(d) NDCG@5 on AppSet100

(e) P@5 on AppSet200

(f) NDCG@5 on AppSet200

Fig. 4. Performance variation with k and λ_L

6.3 Word Vector

To illustrate the learned word vector, we present the latent space with 2-dimension in Fig. 3. Due to the cosine similarity adopted in all the methods, we focus on the angle between two word vectors. Thus all the word vectors are mapped to the unit circle for better comparison. We select two similar app pairs ("tinyCam Monitor FREE","IP Webcam") and ("Strategy & Tactics: WW II FREE","Ice Cream Jump"), where the former means the query app and the latter means the app to be ranked. After preprocessing, there remains three words ("Monitor","IP","Webcam") in the first example, and seven words except "FREE" in the second example.

Both examples in Fig. 3 are similar pairs, so they are supposed to have a small angle between words from different apps. For the first example in Fig. 3a, the query app is for remote surveillance or control your private or public network/IP cameras, video encoders, DVRs and Webcams. For our understanding, both angles between vectors "monitor" and "Webcam", vectors "monitor" and "IP" are small. AppLe-CMF reflects our understanding well while baselines do not. This seems to reveal the underlying reason for the improvement of AppLe-CMF.

6.4 Parameter Sensitivity

There are two parameters in our proposed AppLe-CMF. k is the dimension of the latent space, and λ_L is the trade-off parameter, where $\lambda_E = 1 - \lambda_L$. Here we investigate its sensitivity to different parameter settings. Experimental results are shown in Fig. 4, k ranges from 100 to 1000 with step 100 and λ_L ranges from 0.1 to 0.9 with step 0.1.

In Fig. 4, the relative performance changes with λ_L is stable. For example, on AppSet50, NDCG@5 increases by 1.22 % and P@5 increases by 1.80 % from 0.1 to 0.9 when $k = 1000$. However, the performance increases fast with k from 100 to 1000. For $\lambda_L = 0.9$ on AppSet100, P@5 and NDCG@5 improvement is 9.87 % and 10.59 % respectively. Thus the setting of k has more effect on the performance of our proposed method.

7 Conclusions and Future Work

We mainly focus on learning app representation from app names for similar app recommendation. First, we explore the characteristics of names and find the reason why additional information is needed lies in the discriminative and fewer words in app names. Then we incorporate the word dependency from auxiliary documents, such as app descriptions, into name-word matrix and formulate this app representation learning as the collective matrix factorization problem, referred to as AppLe-CMF. Finally, for better convergence, Alternating direction method of multipliers is employed novelly for AppLe-CMF. Experimental results show that our proposed method outperforms state-of-the-art baselines.

In the future work, we will extend our AppLe-CMF to all the heterogeneous data in apps, such as links, images and other texts, to obtain a comprehensive app representation. Moreover, our AppLe-CMF will be applied to the app search task and how to model query will become of our main concern. Finally, AppLe-CMF in the supervised scenario will be attractive.

Acknowledgments. This paper is supported in part by the National High-tech R&D Program of China (No. 2012AA010902), and NSFC (No. 61303059).

References

1. Kajanan, S., Pervin, N., Datta, A., Dutta, K.: Mobilewalla: a mobile application search engine. IEEE Trans. Mob. Comput. (2011)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. JMLR **3**, 993–1022 (2003)
3. Chen, N., Hoi, S.C., Li, S., Xiao, X.: Simapp: a framework for detecting similar-mobile applications by online kernel learning. In: WSDM 2015, pp. 305–314 (2015)
4. Leung, K.W.T., Ng, W., Di Jiang, K., Vosecky, J.: Panorama: a semantic-aware application search framework. In: Extending Database Technology (2013)
5. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
6. Yang, C.S., Salton, G., Wong, A.: A vector space model for automatic indexing. Commun. ACM **18**(11), 613–620 (1975)
7. Harris, Z.: Distributional structure. Word **10**(23), 146–162 (1954)
8. Shawe-Taylor, J., Cristianini, N., Watkins, C., Lodhi, H., Saunders, C.: Text classification using string kernels. JMLR **2**, 419–444 (2002)
9. Manning, C.D., Pennington, J., Socher, R.: Glove: Global vectors for word representation (2014)
10. Jones, R., Rey, B., Madani, O., Greiner, W.: Generating query substitutions. In: WWW 2006, pp. 387–396 (2006)
11. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
12. Metzler, D., Dumais, S.T., Meek, C.: Similarity measures for short segments of text. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECiR 2007. LNCS, vol. 4425, pp. 16–27. Springer, Heidelberg (2007)
13. Parikh, N., Boyd, S.: Proximal algorithms. Found. Trends Optim. **1**(3), 127–239 (2014)
14. Park, D.H., Liu, M., Zhai, C., Wang, H.: Leveraging user reviews to improve accuracy for mobile app retrieval. In: SIGIR 2015, pp. 533–542 (2015)
15. Peter, P.P., Turney, D.: From frequency to meaning: vector space models of semantics. J. Artif. Intell. Res. **37**(1), 141–188 (2010)
16. Mikolov, T., Le, Q.: Distributed representations of sentences and documents (2014)
17. Rajaraman, A., Ullman, J.D., Leskovec, J.: Mining of Massive Datasets, vol. 1. Cambridge University Press, Cambridge (2012)
18. Sahami, M., Heilman, T.D.: A web-based kernel function for measuring the similarity of short text snippets. In: WWW 2006, pp. 377–386 (2006)
19. Furnas, G.W., Landauer, T.K., Harshman, R., Scott, D., Dumais, S.T.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**(6), 391–407 (1990)
20. Singh, A.P., Gordon, G.J.: Relational learning via collective matrix factorization. In: KDD 2008, pp. 650–658 (2008)

21. Chen, K., Corrado, G.S., Dean, J., Mikolov, T., Sutskever, I.: Distributed representations of words and phrases and their compositionality. In: NIPS (2013)
22. Von Neumann, J.: Functional Operators: The Geometry of Orthogonal Spaces. Princeton University Press, Princeton (1950)
23. Yih, W.-T., Meek, C.: Improving similarity measures for short segments of text. In: AAAI, vol. 7, pp. 1489–1494 (2007)
24. Zhang, Y.: An alternating direction algorithm for nonnegative matrix factorization (2010, preprint)
25. Zhu, H., Chen, E., Xiong, H., Cao, H., Tian, J.: Mobile app classification with enriched contextual information. IEEE Trans. Mob. Comput. **13**(7), 1550–1563 (2014)
26. Zhu, Y., Chen, Y., Lu, Z., Pan, S.J., Xue, G.R., Yu, Y., Yang, Q.: Heterogeneous transfer learning for image classification. In: AAAI (2011)
27. Zhuo, J., Huang, Z., Liu, Y., Kang, Z., Cao, X., Li, M., Jin, L.: Semantic matchingin app search. In: WSDM 2015, pp. 209–210 (2015)

Summarizing Multimedia Content

Natwar Modani[1(✉)], Pranav Maneriker[1], Gaurush Hiranandani[1],
Atanu R. Sinha[1], Utpal[1], Vaishnavi Subramanian[1], and Shivani Gupta[2]

[1] BigData Experience Lab, Adobe Research, Bangalore, India
{nmodani,pmanerik,ghiranan,atr,utp,vasubram}@adobe.com
[2] Adobe India, Noida, India
shivanig@adobe.com

Abstract. Today multimedia content comprising both text and images is growing at a rapid pace. There has been a body of work to summarize text content, but to the best of our knowledge, no method has been developed to summarize multimedia content. We propose two methods for summarizing multimedia content. Our novel approach explicitly recognizes two desirable, normative characteristics of a summary - good coverage and diversity of the respective text and images, and that text and images should be coherent with each other. Two methods are examined - graph based and a modification to the submodular approach. Moreover, we propose a metric to measure the quality of a multimedia summary which captures coverage and diversity of text and images as well as coherence between the text and images in the summary. We experimentally demonstrate that the proposed methods achieve good quality multimedia summaries.

Keywords: Summarization · Text and images · Multimedia content · Algorithms

1 Introduction

Today multimedia content is growing at a rapid pace on the web. To cater to readers, publishers such as The New York Times and The Wall Street Journal offer briefings of content with text and images. The growing shift to mobile devices calls for summarizing multimedia content. Text summarization has been addressed. Our research fills a void by examining summarization of multimedia content - text and images.

The first of two formulations we propose is graph based, inspired by [13]. Each fragment of either content type is a node, the edge weight within a content type is the similarity between two fragments, and the edge weight between fragments of two different types is the coherence between them. The node weight signifies the amount of information in the fragment. The objective function includes all three properties. The second approach uses sub-modular functions, inspired by [9]. The objective function models coverage and diversity of both content types in the summary, but introduces an additional term for the coherence between these types.

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 340–348, 2016.
DOI: 10.1007/978-3-319-48743-4_27

In the absense of ground truth, information coverage and diversity of a summary are used to measure its quality. We extend this notion to images, while also incorporating the coherence of the images and text to define quality for the new concept of multimedia summary. A quality metric, labeled MuSQ (Multimedia Summary Quality) is introduced. With a small manually annotated data set, we demonstrate that the proposed metric shows better agreement with human judgement when compared to traditional metrics such as retention/compression rate and KL divergence. We then evaluate the proposed algorithms using this metric, and the experimental results show that our proposed algorithms perform better compared to the baseline methods.

2 Related Work

While text summarization has been an active area of research for several years, summarizing multimedia content is relatively unexplored. Recent work has presented a multimedia summarizer system for retrieving relevant information from web repositories based on the extraction of semantic descriptors of documents [1]. In this approach, images are not treated as primary objects, but are chosen secondarily based on the selected text summary. Notably, the content of the images is not leveraged, instead only its metadata is used, making the summary potentially less accurate.

The literature on summarization of multimedia data [3] focuses largely on video summarization. Other works [2], based on video/audio features, exploit natural language engines to create textual summaries.

For text summarization, the two broad approaches are: abstractive and extractive. In this research paper, we will be focusing on extractive summarization only.

Starting with Luhn [10] automated (text-only) extractive document summarization has been examined by researchers in Information Retrieval and Computational Linguistics [14]. Algorithms such as support vector machine (SVM) and regression models have been used. However, Wu et al. [17] found that certain graph-based algorithms (for example, TextRank [11]) perform better than SVM and regression methods.

Solving the summarization problem for product reviews, [13] proposed a graph based formulation which uses a fast and scalable greedy algorithm. They considered the informativeness and diversity of the sentences to select the summary of the reviews.

The papers mentioned above follow the bag of words approach, which rely on frequency of words in documents. In a different approach, [5] used continuous vector representations for semantically aware representations of sentences as a basis for measuring similarity. Our technology extends the approach presented in [5,13] to incorporate images in the summary.

With our goal of multimedia summary it is necessary to associate segments of text with segments of images. Approaches that describe contents of images are formulated either by mapping images to a fixed set of human-constructed sentences [4,15], or by automatically generating novel captions [8,12]. Other approaches use Kernel Canonical Correlation Analysis [16] to align images and sentences; however their reliance on computing kernels, quadratic in number of images and sentences, make them not easily scalable. We use the framework developed by [6] to map the text and images onto a common vector space in our work.

3 Problem Definition

First, we present five desirable qualities of multimedia summary qualitatively, by extending well-established concepts in text summarization.

- The text (image) part of the summary should provide good coverage of the text (image) part of the document.
- The text (image) part of the summary should be diverse.
- The text and image part of the summary should be coherent.

We start by defining the content fragment which is either a text unit (typically, a sentence), or an image segment. The desired size of the summary images is a configuration parameter of our system. The image segments are generated as follows. First, we apply an image segmentation algorithm [7] to identify informative objects in an image. Then, each image segment is bounded by a box. This is achieved by finding the smallest rectangle parallel to boundaries that completely encloses the informative object as identified above. If the rectangle is smaller than desired size, it is merged with other image segments that overlap with it. Eventually, when the bounding rectangle is at least of the desired size, we re-size it (by zoom out) to fit the desired size. Now, each such rectangle is an image segment.

The similarity between a pair of text units (sentences) is determined by first applying a recursive auto-encoder based vector representation to both the text units and then taking the cosine similarity between the two vectors. For finding the similarity between a pair of image segments, we apply the deep learning based CNN (convolutional neural network) technique [6] to transform images into a vector of size 4096, and then assess the cosine similarity between these two vectors. To find the similarity between a text unit and an image segment, we apply the transformation to project them into a common vector space [6] and then we compute the cosine similarity between the vectors representing the image and the text.

3.1 Graph Based Approach

In this approach, (inspired by [13]), we construct a graph to represent the document. Each node represents a content fragment. We draw an edge between two nodes, representing two content fragments, with the edge weight as their similarity. We also assign a reward to each content fragment. A text unit is assigned the reward score as the number of nouns, adverbs, adjectives, verbs and half of the number of pronouns. An image fragment is assigned the reward score based on the information content. We take the image segment reward as the average level of similarity with all other image segments.

We attach a cost to each content fragment. The cost of a text fragment is taken in units of sentences, word or characters, and the cost of an image segment is taken as one unit, as all image segments are resized to the desired level. The user also specifies the upper limit on the size of summary for the text and image parts separately, called as budget for the text and image parts, respectively, and represented as b_T and b_I.

We follow an iterative greedy strategy [13] to select the content fragments to include in the summary. In particular, we find the gain G_i of including an available content fragments i in the summary, given by:

$$G_i = \sum_{j=1}^{n} w_{ij} * R_j + \sum_{k=1}^{m} \hat{w}_{ik} * \hat{R}_k \tag{1}$$

Here, w_{ij} is the edge weight between the i^{th} content fragment and j^{th} text unit, and $\hat{w}_{ik}$ is the edge weight between the i^{th} content fragment and k^{th} image segment. Further, R_j is the reward of the j^{th} text unit, and $\hat{R}_k$ is the reward for the k^{th} image segment.

Then we find the content fragment, with the maximum gain to cost ratio, and include it in the summary. Note that we do not impose any order while choosing the text and image fragments for the summary, although the number of text units and image segments selected are controlled by the individual budgets for those two parts of the summary. When a content fragment is included in the summary, the rewards for all other content fragments are updated, per following rules. If a content fragment is the same type as the selected content fragment, its rewards is multiplied by $(1 - w_{ij})$, and if the content fragment in question is of a different type compared to the selected content fragment, its reward is multiplied by $(1 + w_{ij})$. This ensures diversity because the value of including another content fragment that is similar and of the same type as the summary is reduced. At the same time, coherence is achieved since the value of including a content fragment that is similar but of a different type is increased.

3.2 Coverage-Diversity Based Approach

In this approach, inspired by the sub-modular approach to text summarization [5], we have a five part objective function. We have a text coverage term, and a text diversity reward term. Along similar lines, we define the image coverage term, and an image diversity reward term. Finally, we define a coherence term which captures the similarity between text and image(s) selected in the summary. For document D, we denote the summary of the text T as S and of the images V as I. The objective function is defined as

$$F(S, I) = \alpha_1 * C_T(S) + \alpha_2 * R_T(S) + \alpha_3 * C_V(I) + \alpha_4 * R_V(I) + \alpha_5 * H(S, I) \tag{2}$$

Here, α's represent the weights which can be tuned by the user.

The term $C_T(S)$ represents the coverage of the text T of the document by the summary text S, defined in the same way as [5]

$$C_T(S) = \sum_{i \in T} min\{\sum_{j \in S} w_{ij},\ \alpha \sum_{j \in T}\{w_{ij}\}\} \tag{3}$$

The term $R_T(S)$ is the reward for diversity of the text summary S with respect to the text of the document, defined in the same way as [9]

$$R_T(S) = \sum_{i \in S} \sqrt{\sum_{j \in P_i \cap S} r_j} \text{ where } r_j = \frac{1}{n} \sum_{i \in T} w_{ij} \tag{4}$$

where P_i is a partition of the ground set T into separate clusters and r_j is the singleton reward of including sentence j in the empty summary. The clustering is done using CLUTO with the 4096 sized vector representation of the sentences derived from [5] with number of clusters as 0.2 times the number of sentences (so, on average, each cluster would have 5 sentences), a direct K-mean clustering algorithm is used following the same choice as made in [9]. The term r_j is defined again in the same manner as [9] where n is the number of sentences in T and w_{ij} is the similarity between sentences i and j. By replacing T with V and S with I, we can define the corresponding terms for images and their summary.

The term $H(S, I)$ represents the coherence between the summary text and summary images. It is defined as the sum of all pairs of text units and image fragments, i.e.,

$$H(S, I) = \sum_{i \in S} \sum_{j \in I} \hat{w}_{ij}$$

here, $\hat{w}_{ij}$ represents the similarity between the text fragment i in the text part of the summary and image fragment j in the image part of the summary.

4 Multimedia Summary Quality

Measuring quality of a summary relative to its original source is important. Since the problem of multimedia summarization has not been addressed, no quality metrics have been proposed. We propose *MuSQ*, or *Multimedia Summary Quality*, which includes the desirable characteristics stated in Sect. 3. This metric does not require ground truth.

Let the similarity between a content fragment (text or image) u and another content fragment v be given by $Sim(u, v)$. Consider a text sentence v present in the document text T and a sentence u in the summary text S.

Now consider a metric μ_T defined as

$$\mu_T = \sum_{v \in T} R_v * \max_{u \in S} \{Sim(u, v)\} \tag{5}$$

The term $\max_{u \in S} Sim(u, v)$ represents the maximum level of similarity between a sentence v in the document text and any sentence in the summary S.

Recall that the term R_v is the reward value of the sentence v, and contribution of the sentence v towards the quality of the summary is accordingly $R_v * \max_{u \in S} Sim(u, v)$. Note that due to the *max* function, if there are two sentences which are similar to the given sentence v, it will not lead to enhanced contribution of the sentence to the quality of the summary. On the other hand, if the summary is having a sentence similar to a sentence in the document, it leads to increase in the metric value for the summary quality. In this way, the function μ_T is able to simultaneously capture the diversity and the information content of the summary with respect to the text of the original document T.

We define the overall quality metric *MuSQ* as:

$$\mu_M = \mu_T + \mu_I + \sigma_{T,I} \tag{6}$$

$$\mu_I = \sum_{w \in V} \hat{R}_w * \max_{x \in I} \{Sim(w, x)\} \tag{7}$$

$$\sigma_{T,I} = \sum_{v \in S} \sum_{w \in I} \{Sim(v, w) * R_v * \hat{R}_w\} \tag{8}$$

The terms μ_T and μ_I are diversity aware information coverage measure for the text part and the image part of the summary, respectively. The third term $\sigma_{T,I}$ measures the degree of cohesion between the text and the image part of the summary, as the sum of similarities between the sentences and the images in the summary, across all pairs.

5 Experimental Results

Now, we describe experimental results to validate our algorithms, as well as the proposed quality metric. First, on a small dataset we check whether the quality metric *MuSQ* correlates well with human judgment about the quality of multimedia summary, since obtaining human input for a large dataset is very expensive. Once *MuSQ* is validated, it is used to evaluate the proposed summarization algorithms on a larger dataset.

The small dataset comprised ten articles from the *New York Times* for each of which we created two summaries. In a survey, participants were shown the original article, the two summaries and were asked which one of the two summaries was better, or whether they were almost of similar quality. To control the order effect, the summaries were randomly placed first or second (without regard to their *MuSQ* scores), and the participants were not given any information about how the summaries were generated.

We define agreement level in three different ways. The first definition treats the 'Equal' option as half agreement and half disagreement, i.e., $AL1 = 100 * (A+0.5E)/(A+E+D)$, where $AL1$ is the agreement level according to definition 1, A is number of agreements (i.e., the human judge preferred the summary which had higher *MuSQ* score), E is number of times both summaries were deemed to be of same quality by the human judge, and D is the number of disagreements (i.e., the human judge preferred the summary which had lower *MuSQ* score).

Second definition treats the 'Equal' option as disagreement, i.e., $AL2 = 100 * A/(A + E + D)$. Third definition ignores the 'Equal' option completely, i.e., $AL3 = 100 * A/(A + D)$.

In total, 22 human judges provided 128 responses. Out of these, 87 responses favoured summaries with higher *MuSQ* scores, whereas 14 responses found the summaries to be almost equal in quality. The remaining 27 responses disagreed with the ranking based on the *MuSQ* scores. This translates to 68 % agreement for the *MuSQ* scores (where, as a conservative approach 'equal' is classified as a disagreement), and 76 % agreement ignoring the votes for 'equal'. The Pearson correlation coefficient between the agreement levels (AL1, AL2 and AL3) and the fractional difference in the *MuSQ* scores is approximately 0.51 for all the three definitions of agreement levels, which shows that our proposed quality metric correlates well with human judgment.

Now, we describe the experiments performed on a larger dataset, considering *MuSQ* as the quality metric. We collected 1, 000 articles from *New York Times*, which typically have text and images, both. We kept only those articles which had at least 20 and at most 100 sentences, and at least 1 image. This resulted in selecting 703 articles for the experiment. Further, the size of the summary was specified as 3 sentences and 1 image of size $200 * 200$ pixels.

The image segmentation algorithm takes the number of objects to be identified as input. We choose to identify 20 objects, with a further constraint that each class of objects does not occur more than 10 times. This ensures that the objects from a general class, such as background, do not end up as the only objects in the segments. Also, we used [9] to compute the similarity between two sentences. The similarity between two images, as well as, between a text sentence and an image was computed in the same way as [6].

We evaluated the two approaches proposed in this paper using the *MuSQ* score. As a baseline, we used the text only version of these two algorithms for finding the three summary sentences, and augmented this summary with the first (whole) image from the article (hitherto only known method). The graph based approach we propose achieves the highest score 539 times, and the coverage-diversity based approach achieves the highest score 90 times. Only 103 times out of 703 articles, one of the two baseline approaches outperform our proposed approaches, and 587 times our proposed approaches outperform the baseline approaches. This means that our proposed approaches are better 83.5 % of the times and equally good another 1.5 % of the times. As the *MuSQ* scores are dependent on the size of the original document, it is not appropriate to compare them across articles.

We also report the traditional text only performance metric for the summary quality for the four algorithms in Table 1, as well as the newly proposed metric *MuSQ*. As expected, the *MuSQ* score is higher for the enhanced versions compared to the baseline methods. One finds that both for retention rate and KL-Divergence, the baseline approaches perform better than the enhanced approaches, which are to be expected. However, note that the performance degradation is fairly small and less severe for the graph based approach. Hence, the algorithms proposed by us provide significant value for summarizing multimedia content.

Table 1. Quality metric for the four approaches (retention rate and KL-Divergence are measured only for the text part of the summary)

Metric	Enhanced approaches		Baseline approaches	
	Submodular	Graph based	Submodular	Graph based
MuSQ	1528.37	1592.18	1519.95	1564.78
Retention rate	0.3704	0.4608	0.3896	0.4652
KL-Divergence	1.2052	0.8980	1.0822	0.8725

6 Conclusion

Today multimedia content in the form of text and images are commonplace across publishing sites and devices. The need for the summarization of such content to comprise both text and images is stronger than ever before. The results provide strong evidence in support of our proposed methods and validate the new quality metric. These summaries are better than the summaries generated only using text part and then adding the first image, which is the only known multimedia summary method. We hope that future work will advance our understanding and knowledge in multimedia summarization to parallel that of text summarization.

References

1. dAcierno, A., Gargiulo, F., Moscato, V., Penta, A., Persia, F., Picariello, A., Sansone, C., Sperl, G.: A multimedia summarizer integrating text and images. In: Intelligent Interactive Multimedia Systems and Services, pp. 21–33. Smart Innovation, Systems and Technologies (2014)
2. Ding, D., Metze, F., Rawat, S., Schulam, P.F., Burger, S.: Generating natural language summaries for multimedia. In: Proceedings of the Seventh International Natural Language Generation Conference, pp. 128–130. Association for Computational Linguistics (2012)
3. Ding, D., Metze, F., Rawat, S., Schulam, P.F., Burger, S., Younessian, E., Bao, L., Christel, M.G., Hauptmann, A.: Beyond audio and video retrieval: towards multimedia summarization. In: Proceedings of the 2nd ACM International Conference on Multimedia Retrieval, p. 2. ACM (2012)
4. Farhadi, A., Hejrati, M., Sadeghi, M.A., Young, P., Rashtchian, C., Hockenmaier, J., Forsyth, D.: Every picture tells a story: generating sentences from images. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 15–29. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15561-1_2
5. Kageback, M., Mogren, O., Tahmasebi, N., Dubhashi, D.: Extractive summarization using continuous vector space models. In: Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC), pp. 31–39. EACL (2014)
6. Karpathy, A., Joulin, A., Fei-Fei, L.: Deep fragment embeddings for bidirectional image sentence mapping. Archive, Cornell University Library (2014). http://arXiv.org/abs/1406.5679

7. Krähenbühl, P., Koltun, V.: Geodesic object proposals. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 725–739. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10602-1_47
8. Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A.C., Berg, T.L.: Baby talk: understanding and generating simple image descriptions. In: CVPR (2011)
9. Lin, H., Bilmes, J.: A class of submodular functions for document summarization. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, HLT 2011, Stroudsburg, PA, USA, vol. 1, pp. 510–520 (2011)
10. Luhn, H.: The automatic creation of literature abstracts. IBM J. Res. Dev. **2**(2), 159–165 (1958)
11. Mihalcea, R.: Language independent extractive summarization. In: ACLdemo, pp. 49–52 (2005)
12. Mitchell, M., Han, X., Dodge, J., Mensch, A., Goyal, A., Berg, A., Yamaguchi, K., Berg, T., Stratos, K., Hal Daum, I.: Midge: generating image descriptions from computer vision detections. In: EACL (2012)
13. Modani, N., Khabiri, E., Srinivasan, H., Caverlee, J.: Graph based modeling for product review summarization. In: WISE (2015)
14. Nenkova, A., McKeown, K.: A survey of text summarization techniques. In: Aggarwal, C.C., Zhai, C.X. (eds.) Mining Text Data, pp. 43–76. Springer, New York (2012)
15. Ordonez, V., Kulkarni, G., Berg, T.L.: Im2text: describing images using 1 million captioned photographs. In: NIPS (2011)
16. Socher, R., Fei-Fei, L.: Connecting modalities: semi-supervised segmentation and annotation of images using unaligned text corpora. In: CVPR (2010)
17. Wu, J., Xu, B., Li, S.: An unsupervised approach to rank product reviews. In: FSKD, pp. 1769–1772 (2011)

Bridging Enterprise and Software Engineering Through an User-Centered Design Perspective

Pedro Valente[1,2(✉)], Thiago Silva[1], Marco Winckler[1],
and Nuno Nunes[2]

[1] Institut de Recherche en Informatique de Toulouse (IRIT),
Université Paul Sabatier, Route de Narbonne, 118, 31400 Toulouse, France
pvalente@uma.pt, {rocha,winckler}@irit.fr
[2] Madeira Interactive Technologies Institute (MITI), University of Madeira,
Caminho da Penteada, 9020-105 Funchal, Portugal
njn@uma.pt

Abstract. The development of Web-based Information Systems is crucial in the quest to maintain and develop the enterprise competiveness. However, capturing requirements from Business Processes (BP) is still an issue, as existing methods mostly focus, or on human aspects and the user interface, or on business concerns as rules and workflow coordination, and therefore do not specify all the Software Architectural components which are relevant for software development. We present the Goals Approach, which analyzes BPs and User Tasks and details them in the process of methodically designing and structuring the User Interface, the Business Logic and the Database of the Information System given a Model-View-Controller (MVC) architectural pattern. In this paper we focus on how to obtain the Goals business model of requirements based on the DEMO method. The approach can be used for in-house software development, and the method is straightforward fitting Small and Medium Enterprises agility needs.

Keywords: Web-based applications · Enterprise engineering · Software engineering · User-Centered Design · Software architecture

1 Introduction

Software development within enterprises still lacks effectiveness as project full-success rates are still as low as about 30 % [1, 2]. Despite this fact, efforts in SE have at least taken us from a chaotic state of the practice [3], to a more inspiring situation where expertized executive management support, user involvement in the development process and agile techniques are appointed as factors of project success [4, 5].

In our quest to integrate the enterprise and the software engineering perspectives as a solution to align business and Information Technology (IT) and create the conditions to increase software success rates, we bridge both domains by means of a User-Centered Design perspective that allows the modeling of Web-based applications. We presents the Goals Approach, which models the business and uses it as the back-bone of the software architecture. This paper focuses on the business model elaboration from DEMO [6], as a way to enhance business analysis performance and explain our method.

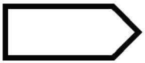
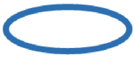
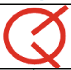
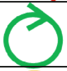
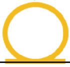
W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 349–357, 2016.
DOI: 10.1007/978-3-319-48743-4_28

1.1 Foundations and Software Development Process

The Goals Approach is founded on five methods: Wisdom [8], which is a software engineering and architectural method; Goals [9], which establishes a relation between business and software architectures; DEMO [6], that models the enterprise by means of an ontology; Activity Modeling (AM) [10], which models human activity and designs the user interface, and BDD [11], which models user interface and system behavior.

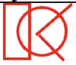
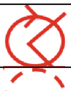
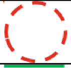
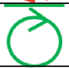
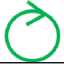
The Software Development Process defines a method that integrates the Enterprise Engineering and Software Engineering perspectives, concerning a given Business Process Improvement (BPI) [12], in two phases. The Analysis Phase identifies Business Processes (Step 1), User Tasks (Step 2), Interaction Spaces (Step 3), Business Rules (Step 4) and Data Entities (Step 5), composing an Enterprise Structure of business requirements, which components are presented in Table 1.

Table 1. Enterprise structure's component's definition, origin and symbol.

Component	Brief Definition	Origin	Symbol
Business Process (BP)	*A Network of UTs that lead to a Goal*	DEMO	
User Task (UT)	*A Complete Task within a BP*	AM	
Interaction Space (IS)	*The Space that supports a UT with the same BRs and DEs.*	Wisdom	
Business Rule (BR)	*A Restriction on the DE's Structural Relations*	DEMO	
Data Entity (DE)	*Persistent Information about a Business Concept*	Wisdom	

The Design Phase applies a User-Centered Design perspective to the Enterprise Structure in order to specify User Tasks (Step 6), design the User Interface (Step 7), structure the Business Logic (Step 8) and the Database (Step 9), finishing (Step 10) by elaborating the Software Architecture based on a MVC architectural pattern [13] in order to support for any possible combination of BPs that may structure the enterprise service.

Table 2. Software Architecture components definition, origin and symbol.

Component	Definition	Origin	Symbol
Aggregation Space	*A User Interface*	Hydra	
Interaction Component	*Tool of a User Interface*	Goals	
Interaction Object	*A User Interface Object that triggers SRs*	Goals	
User Interface SR	*A SR that provides support for User Interface presentation*	Goals	
Database SR	*A SR that manages Data Entities*	Goals	

Each Software Architecture component is presented in Table 2, where SR stands for System Responsibility. The Software Architecture is elaborated by means of composing one Aggregation Spaces [14] per each User Task (UT), which architecturally uses the ISs which are associated to the UT, ensuring the application of BRs over identified DEs and ensuring traceability between business and software implementation.

Goals establishes a relation with DEMO by means of the concepts of BP, UT, BR and DE which are compatible with the DEMO concepts of Transaction, Coordination Act, Action Rule and Object Class, respectively. Goals adds the Interaction Space (IS) which as the key to build-up the Enterprise Structure. We define three patterns of derivation (A, B and C) which are used to identify Goals component from DEMO models. The patterns are introduced in Fig. 1, and Steps 1 to 5 which explain the derivation of components in a top-down process are presented in Sects. 2.1, 2.2, 2.3, 2.4 and 2.5.

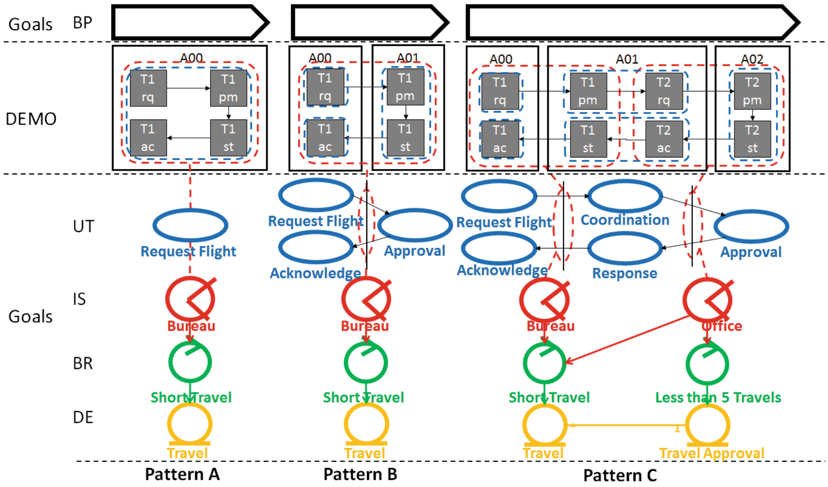


Fig. 1. Patterns A, B and C of component (BP, UT, IS, BR and DE) derivation from DEMO.

2 Analysis Phase

The elaboration of the Enterprise Structure is presented in Steps 1 to 5.

2.1 Step 1 – Business Process (BP) Identification

Goals definition of BP is compliant with the notion of Business Process provided by DEMO as a “set of interrelated or enclosed Transactions”. One Transaction is a sequence of Coordination Acts {namely: request (rq), promise (pm), state (st) and accept (ac)} performed by two actors, or by a single actor directly in the system.

Figure 1 presents the BP derivation patterns based on the DEMO Process Structure Diagram (PSD). Pattern A includes a single Transaction (T1) performed by Actor A00, pattern B includes a single Transaction (T1) performed by two Actors (A00 and A01), and pattern C has two Transactions (T1 and T2) performed by three Actors (A00, A01 and A02). In all cases the relation between Goals and DEMO BPs is of one-to-one.

2.2 Step 2 – User Task (UT) Identification

Contrarily to DEMO, Goals considers that an Actor always carries on a only single task (a UT) and never two consecutive tasks or Coordination Acts (C-Acts). This aims Business Process clarification, user performance and software conception efficiency in order to deploy the necessary tools for the execution of the task by reducing articulatory distance and therefore, the user effort [15]. Hence, Goals considers any consecutive DEMO C-Acts {request (rq), promise (pm), state (st) and accept (ac)] as a single UT.

Figure 1 presents the derivation of UTs from the DEMO PSD. In pattern A a single UT is considered for the four consecutive C-Acts {rq, pm, st and ac}. In pattern B the consecutive C-Acts {pm and st} performed by Actor A01 are considered as a single UT ("Request Flight"), and in pattern C, Actor A01 is responsible for transposing the BP execution from Actor A00 to A02 and viceversa by carrying on the UTs "Coordination" and "Response", which are merged from consecutive C-Acts, namely {T1 pm - T2 rq} and {T2 ac - T1 st} respectively.

2.3 Step 3 – Interaction Space (IS) Identification

One IS supports the interaction between two users in person or remotely while each one carries on his own UT. Even if many UTs are carried by many Actors, the UTs will still be different, and if two Actos carry on the same UT of the BP remotely, then they are performing cooperative work [16]. The derivation of ISs does not depend on DEMO models as this method does not consider the space where human activity occurs.

Figure 1 illustrates the derivation of the IS from the relation of UTs, as each IS (e.g. "Bureau" in Pattern B) supports the communication between any two or more Actors, the line that divides the swim-lanes of each Actor represents an IS. Given that DEMO only predicts the interaction between two Actors, when applied to DEMO models, this pattern of derivation will always result in a direct relation between one IS (e.g. ISs "Bureau" and "Office" for Transactions T1 and T2 in Pattern C) per Transaction.

2.4 Step 4 – Business Rule (BR) Identification

BRs represent regulations or requirements that should be elicited during the Analysis Phase in order to facilitate the understanding of the restrictions which the user is subject to when carrying on a User Task within a certain Interaction Space, and represent restrictions which are applied to existing Data Entities. BRs are the grounding foundation of the Business Logic (given an MVC pattern), as they are the more specific programmed system responsibility regarding the structuring of this layer, the middleware of the system.

Figure 1 illustrates a situation in which both T1 and T2 define a BR each ("Short Travel" and "Less than 5 Travels"), which are also used by the Interaction Spaces of Transactions T1 and T2. BRs are constantly executed in order to ensure that a given restriction is ensured regarding the transfer of information between Interaction Spaces and Data Entities. BR should also be used by Interaction Spaces in order to restrict the introduction of invalid information by the user, therefore preventing usage mistakes.

2.5 Step 5 – Data Entity (DE) Identification

DEs are business concepts which are recognized within the enterprise domain by those who have knowledge about it (the enteprise). DEs are compliant with the concept of Class and relation of Classes used in UML [17]. And this definition is compatible with DEMO Object Classes (OC) which structures facts and Transactions. Goals derivation of DEs is carried out using the Object Fact Diagram (OFD) by establishing a direct relation between one DE per OC.

Figure 2 which presents the relation between OCs and Transactions in the OFD horizontal swim-lane. Transactions 1 and 2 use each a single DE, and these are related in a multiplicity of 1 to many (from "Travel" to "Travel Approval"). The resulting Enterprise Structure is presented above the DEs representation, and is composed by every identified component until this moment with no changes and is representative of the social interaction in terms of stable and essential norms, which is known as the organizational kernel [18].

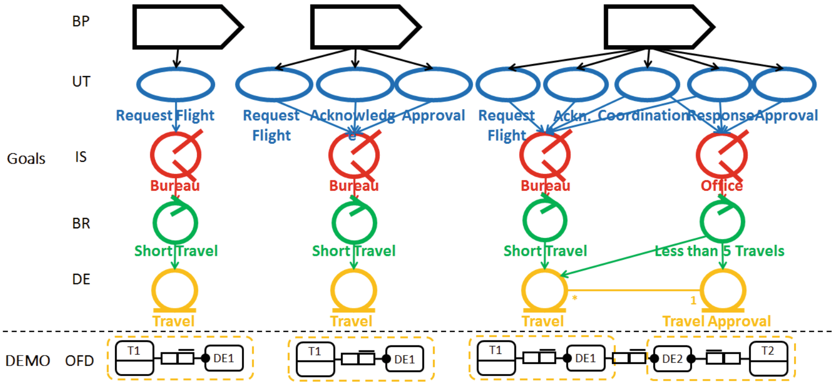


Fig. 2. Enterprise structure and derivation of DEs from OFD diagram.

3 Analysis Phase

The Design Phase elaborates the Software Architecture which is conceived in a top-down process that detailing the User Interaction (Step 6), the User Interface (Step 7), the Business Logic (Step 8) and the Database layer (Step 9), finishes with the composition of the Software Architecture (Step 10).

3.1 Step 6 – Task Model

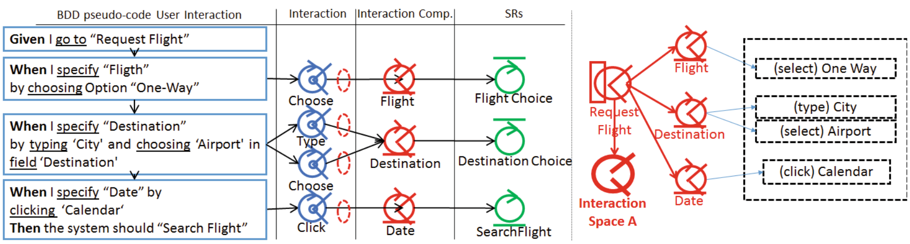
The Task Model details User Tasks (UTs) in order to obtain information in order to carry on the User Interface design, which happens in Step 7. The Task Model follows the technique applied in the Wisdom methodology in order to specify the UT in terms of User Intentions (steps that the user takes to complete the task) and System Responsibilities (that provide the necessary information), following a traditional decomposition of an Essential Use Case (EUC) by means of the application of the Concur Task Trees (CTT) technique [19].

3.2 Step 7 – Interaction Modeling

The User Interface Design is carried out by means of the application of the Behavior Driven Development (BDD) method [11]. BDD is an agile software development method that produces pseudo-code as User Stories in order to specify a system feature (a UT) which is used within a certain scenario (an Aggregation Space).

User Stories specify a flow of user interactions that matches the User Intentions of the Task Model, specifying one Interaction Components per User Intention, and one Interaction Object per User Interaction, and related system behavior in terms of User Interface and Database System Responsibilities (SRs).

Figure 3 presents a User Story example for User Task “Request Flight” where three Interaction Components (A, B and C) and three SRs (the last SR is always a Database SR) are identified following the specification of four User Interactions. The User Interface Design specifies the Aggregation Space which is composed by the Interaction Components and the Interaction Objects (one to support each User Interaction).



Given I go to “Request Flight”

When I specify “Fligth” by choosing Option “One-Way”

When I specify “Destination” by typing ‘City’ and choosing ‘Airport’ in field ‘Destination’

When I specify “Date” by clicking ‘Calendar‘ **Then** the system should “Search Flight”

Fig. 3. Interaction model and user interface example.

3.3 Step 8 – Business Logic Structuring

The Business Logic Structuring is carried out by defining the relations that each System Responsibility (SR) to the existing to Data Entities (DE) based on the semantics and current state about identified business concepts. Given the current example and given Pattern A of derivation, we assume that DEs "Travel" and "Approval" are inherited from the Enterprise Structure. "Flight Choice" has been mapped to "Travel", and it is assumed that the "Airport" Fields belongs to a new DE "Airport". By means of the analysis of "SearchFlight", we assume that it uses a new DE "Flight".

3.4 Step 9 – Database Structuring

The Database Structuring is possible once all new DEs and Fields are already identified. The structuring od carried out according to the principles of elaboration of a Domain Model [17], in terms of Classes and Attributes which suffer simple transformation in order to structure the final Database [20]. According to our example, two new DEs have been identified ("Flight" and "Airport"), and for purposes of exemplification, we assume that DE "Travel" can only be related to a single record of those new DEs, and that DE "Flight" can is related to more than one "Airport" (usually two).

3.5 Step 10 – Software Architecture Composition

The composition of the Software Architecture is carried out by relating in a single diagram the every identified component by means of the execution of Steps 1 to 9. Figure 12 presents the Software Architecture, which relates all the identified components in a single Software Architecture, including the User Interface components associated to UT "Request Flight", the elaborated Business Logic and Database components, including the components which are architectural inherited from the Enterprise Structure.

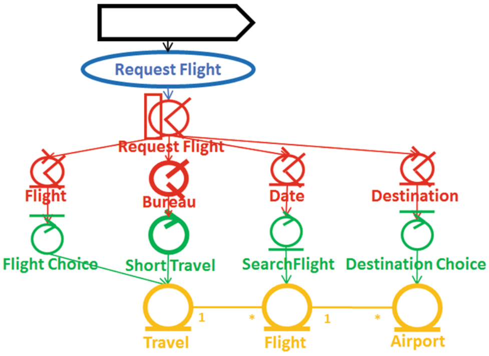


Fig. 3. Software Architecture example.

4 Related Work

Our approach can be compared to ArchiMate [21] and BPMN [22] in the perspective that it provides an Enterprise and Software Structuring language. It is different in the perspective that it applies a methodology to derive software implementation specifications. Regarding the specific User-Centered Design perspective, the closest solutions are methods settle for user interface conception based on user task and domain models, such as Sukaviriya's [23] and Sousa's [24]. Our approach is different as it complementarily conceives the Business Logic layer based on enterprise business rules and coordination structures that operate the user interface and domain processing execution. Considering the enterprise-driven development, it is different from the DEMO-based GSDP [25] as it specifies a structured user interface.

5 Conclusions

Our approach inherently aims at facilitating requirements elicitation, focuses on user needs, and simplifies traceability between business requirements and software implementation, witch match project management needs and user involvement in the SDP. The Goals Approach strategy, which is based on BPI, fits most successfully sized projects. Based on Standish Group statistical reports, projects under 1 M$ (one million dollars) cost are believed to be up to 10 times more successful than 10 M$ projects [4]. It suits Small and Medium Enterprises (SME) in-house development needs of agility concerning the achievement of tangible results in limited amounts of time [7].

References

1. The Standish Group: Chaos Report 2014 (2014)
2. Valente, P., Aveiro, D., Nunes, N.: Improving software design decisions towards enhanced return of investment. In: Proceedings ICEIS 2015, pp. 388–394 (2015)
3. Morgenshtern, O., Raz, T., Dvir, D.: Factors affecting duration and effort estimation errors in software development projects. IST **49**, 827–837 (2007)
4. The Standish Group: Chaos Report 2013 (2013)
5. Version One. The 10th Annual State of Agile Report (2016)
6. Dietz, J.: Enterprise Ontology - Theory and Methodology. Springer, Berlin (2006). ISBN 978-3540331490
7. Gerogiannis, V., Kakarontzas, G., Anthopoulos, L., Bibi, S., Stamelos, I.: The SPRINT-SMEs. In: Proceedings of ARCHIMEDES III (2013)
8. Nunes, N.: Object modeling for user-centered development and user interface design: the wisdom approach. Ph.D. thesis, Universidade da Madeira (2001)
9. Valente, P.: Goals Software Construction Process: Goal-Oriented Software Development. VDM Verlag Dr. Müller, Germany (2009). ISBN 978-3639212426
10. Constantine, L.: Human Activity Modeling - Toward a Pragmatic Integration of Activity Theory and Usage-Centered Design. Springer, Berlin (2009)

11. Chelimsky, D., Astels, D., Helmkamp, B., North, D., Dennis, Z., Hellesoy, A.: The Rspec Book (2010). ISBN: 1934356379
12. Lodhi, A., Köppen, V., Saake, G.: Business process improvement framework and representational support. In: Proceedings of the 3rd International Conference on Intelligent IHCI (2011)
13. Zukowski, J.: The model-view-controller architecture. In: John Zukowski's Definitive Guide to Swing for Java 2 (1999). ISBN: 978-1430252511
14. Costa, D., Nóbrega, L., Jardim Nunes, N.: An MDA approach for generating web interfaces with UML ConcurTaskTrees and canonical abstract prototypes. In: Coninx, K., Luyten, K., Schneider, K.A. (eds.) TAMODIA 2006. LNCS, vol. 4385, pp. 137–152. Springer, Heidelberg (2007)
15. Winckler, M., Cava, R., Barboni, E., Palanque, P., Freitas, C.: Usability aspects of the inside-in approach for ancillary search tasks on the web. In: Abascal, J., Barbosa, S., Fetter, M., Gross, T., Palanque, P., Winckler, M. (eds.) INTERACT 2015. LNCS, vol. 9299, pp. 211–230. Springer, Heidelberg (2015). doi:10.1007/978-3-319-22668-2_18
16. Grudin, J.: Computer-supported cooperative work: history and focus. Computer **27**, 19–26 (1994)
17. Booch, G., Jacobson, I., Rumbaugh, J.: The Unified Modeling Language Users Guide. Addison-Wesley, Wokingham (1998)
18. Stamper, R.: On developing organisational semiotics as an empirical science: the need for scientific method and rigorous debate. In: Proceedings of 14th ICISO, pp. 1–13 (2013)
19. Paternò, F.: Model-Based Design and Evaluation of Interactive Applications. Springer, London (1999)
20. Awang, M., Labadu, N.: Transforming object oriented data model to relational data model. New Comput. Archit. Appl. **2**(3), 402–409 (2012)
21. Archimate Foundation: Archimate Made Practical (2008)
22. Völzer, H.: An overview of BPMN 2.0 and its potential use. In: Mendling, J., Weidlich, M., Weske, M. (eds.) BPMN 2010. LNBIP, vol. 67, pp. 14–15. Springer, Heidelberg (2010). doi:10.1007/978-3-642-16298-5_3
23. Sukaviriya, N., Sinha, V., Ramachandra, T., Mani, S., Stolze, M.: User-centered design and business process modeling: cross road in rapid prototyping tools. In: Baranauskas, C., Abascal, J., Barbosa, S.D.J. (eds.) INTERACT 2007. LNCS, vol. 4662, pp. 165–178. Springer, Heidelberg (2007)
24. Sousa, K., Mendonça, H., Vanderdonckt, J., Rogier, E., Vandermeulen, J.: User interface derivation from business processes: a model-driven approach for organizational engineering. In: Proceedings of 2008 ACM SAC, pp. 553–560 (2008)
25. Kervel, S., Dietz, J, Hintzen, J., Meeuwen, T., Zijlstra, B.: Enterprise ontology driven software engineering. In: Proceedings of ICsoft 2012 (2012)

Special Session on Data Quality and Trust in Big Data

Region Profile Based Geo-Spatial Analytic Search

Xiaofeng Du(✉) and Zhan Cui

British Telecom, London, UK
xiaofeng.du@bt.com

Abstract. In geo-spatial related data analysis, an important task of geo-data analysts is to quickly find the things that they are interested from their data, such as spatial-temporal patterns, spatial clusters, co-location patterns, and spatial hotspots etc. Currently, most of the geo data analytic tools are exploratory based and lack of facilities that can help analysts to define what they are looking for and quickly find them from data. In this paper, we proposed a region profile based geo-spatial data analytic solution that tackles exactly the issue. The proposed solution captures analysts' interests in so called region profiles and then uses those region profiles to quickly locate the data that satisfy those interests either manually or automatically. Through the proposed solution, analysts can easily find what they are looking for in their data. They also can validate their results in a collaborative analytic environment and share and reproduce analytic results across a group of analysts.

Keywords: GIS · Geo-spatial data analytics · Spatial analysis · Point pattern analysis · Spatial statistics

1 Introduction

As the generation of information proliferates, vast quantities of data are created by systems, software, devices, sensors and all manner of other entities. Some data is intended for human review, problem identification or diagnosis, scanning, parsing or mining. As data sets are generated and stored in greater quantities, at greater rates, and with potentially greater levels of complexity and detail, the "big data" problem of storing, handling, processing or using the data arises.

Data from sensors, recorded events, astronomical, medical, computer network and other data can have associated spatial attributes such as geo-spatial coordinates or other relative or absolute spatial location information. For example, traffic, crime, health and social data can be associated with spatial locations in a map. Network data can be associated with real or virtual spatial locations in a network arrangement or topology. Furthermore, data can be associated with temporal attributes such as a measure of an absolute or relative time of occurrence at a relevant degree of granularity and precision. Spatial and spatiotemporal data analysis techniques are known in the art for the analysis of data sets by identifying co-occurrences of data events, similarity of data events and rules or models of data events [6]. For example, in genetics, spatiotemporal gene expression is employed to identify patterns of gene expression [9]. On another scale,

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 361–370, 2016.
DOI: 10.1007/978-3-319-48743-4_29

epidemiological data can be analysed using similar techniques to observe rules relating data events, such as socio-economic events or factors and their association with health conditions [3]. In crime recording, spatiotemporal analysis techniques can identify relationships between criminal occurrences and proximity of certain resources, facilities and times of day, season etc. A variety of analytic techniques and functions can be employed, e.g. spatial or spatiotemporal clustering; co-location pattern analysis; spatiotemporal pattern analysis.

While a variety of analytic techniques and functions exist, their application to potentially large data sets across multi-dimensional spaces such as two-dimensional geographic areas involves considerable analytical effort. Typically, the process of data analytics involves gathering data, loading data into analytic tools, and applying exploratory based statistical models or data mining methods. Through this process, something potentially useful could be discovered, such as clusters, rules, and patterns. However, such an approach is slow and resource intensive and prone to fail to identify analytical results that the analysts are most interested in, especially in the Big Data era with large amount of data coming in at high velocity and rapid changing of analysts' interests. To mitigate these challenges, a system with better utilities that can help geo-data analysts to easily identify interesting information from their data is needed.

We believe that exploratory based data analytic [1, 4, 6–8] only works most effectively at the very beginning stage of the data analytic process, i.e. when an analyst receives a new set of data and needs to be familiar with it. At the later stages, usually the analysts will know what they are looking for in the data, but do not have effective ways of describing and searching for it. At this stage, very often the questions they ask are "the crime pattern I discovered in this city, does it happen in other cities as well?" or "in this area, my products have good sales in the places where there is a retirement home nearby, is this true in other areas too or just a coincident?". For these questions, the analysts need quick answers rather than go through another complex analytic process. In this paper, we propose a system that employs a flexible region profile based geo-spatial analytic search method that can not only discover something interesting from data, but also help analysts to encapsulate the similar questions mentioned above as region profiles and find the data that produce their desired analytic results. The key contributions of our work is summarised as follows:

- **Developing the concept of region profile**. A region profile is a capsule that encapsulates desired analytic results that analysts are looking for and how the results are produced. A region profile is a template spatial region having a shape and dimensions, an identification of a plurality of analytic functions, a specification of one or more parameters for each identified function, and a result set including output of each of the functions applied to events associated with a spatial region of the space in accordance with the template and based on the parameters. More details of profile will be discussed in later sections.
- **Developing an interactive visual analytics environment for users to create/modify region profiles**. Under this environment, users can highlight their interested geo-regions on map, perform statistical and spatial-data mining analytics to get insights into their data in the highlighted regions, and save their desired

results as region profiles for further analysis later. The region profile can be easily shared with other analysts.

- **Developing an interactive search facility for users to quickly locate subsets of data that match pre-defined region profiles.** The search facility can search for subsets of data that match to the profile either automatically or manually controlled by users.

This paper is structured as follows. Section 2 reviews related work. After give the details of what a region profile is in Sect. 3, we will discuss how region profiles can be used in geo-data analytics in Sect. 4. Finally, we discuss the limitations and future work in Sect. 5.

2 Related Work

In this section, we review some of the geospatial data analytic tools that are most related to our work. We will also discuss why our solution is different.

ArcGIS [1] is one of the most comprehensive GIS tool. It provides the system for editing, storing, visualizing and analysing geographic data. Spatial data analysis is one of its strength. It supports extendable analytic capabilities through extensions. However, it is an exploratory based data analytic tool and does not have any functionality that supports users to capture what they are looking for and search through data based on it. Chen et al. [8] developed a system that focuses specifically on discovering people movement pattern through social media data. It allows users to filter and select reliable data from each derived movement category, based on the guidance of uncertainty model and interactive selection tools. By iteratively analysing filtered movements, users can explore the semantics of movements, including the transportation methods, frequent visiting sequences and keyword descriptions. It does mention pattern which has some similarity to region profile, but it is limited to movement analysis only. GeoVISTA Studio [4] is an open source Java-based visual programming environment and is commonly used for developing geo-visualization applications. Another general system is QGIS [7]. It is an open-source desktop GIS application, which supports exploratory data analysis and data editing.

3 Region Profile

As discussed previously, when data analytics progress to certain stages, the analysts should approximately know what they are looking for in their data. The rest of the analytic tasks should be focusing on looking for which part of the data contains their desired interests. This is especially true in geo-spatial data analytics due to spatial data's spatial dependence [5] and spatial heterogeneity [10] characteristics. Once analysts discover some interesting result, they need to validate whether the result holds true in other geographical locations or just a local phenomenon. Traditionally, to validate the result, the analyst can simply load data from other geo regions into the model he/she built to see if they generate the same results because all the data are local and the analyst does not work collaboratively. However, this solution has some major drawbacks, especially in the Big Data era with a group of analyst performing data analytics simultaneously:

- **Low efficiency**. The model needs to go through the whole data set in order to find interests in other areas. As there is no clearly defined "local area", the model also needs to figure out where to start and when to stop the analysis to find interests in other areas and at the same time does not lose data locality.
- **Insufficient privileges to access required data**. In the modern data analytics, especially with the emerging of Big Data, data set are getting extremely large and distributed. More often than not, an analyst does not have access to all the data that he/she needs to validate the results.
- **High complexity and error prone to reproduce results**. It takes considerable amount of efforts and analytical knowledge to share a pre-designed model among analysts and produce the same results. It is even more difficult to share the model with people who have no analytical knowledge.

To overcome the above issues, we propose the concept of region profile. A region profile is a capsule that encapsulates desired analytic results that analysts are looking for and how the results are produced. There are few key elements that are essential in a region profile, such as a clear defined region, a time window, and the analytic functions that are used etc. formally a region profile is defined as follows:

Definition 3.1. A region profile $\boldsymbol{RP}$ is a tuple, $\boldsymbol{RP} = (\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{D}, \boldsymbol{F}, \boldsymbol{P}, \boldsymbol{R})$, where:

- $\boldsymbol{S} = ((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$ is a sequence of coordinates of a polygon that defines a local region on map. It can be a predefined region, such as a shape of a city boundary, or a hand drawn shape, such as a circle.
- $\boldsymbol{T} = [t_1, t_2]$ is a time interval between t_1 and t_2.
- $\boldsymbol{D} = \{d_1, d_2, \ldots, d_n\}$ is a set of data that used to produce the results.
- $\boldsymbol{F} = \{f_1, f_2, \ldots, f_n\}$ is a set of analytic functions.
- $\boldsymbol{P} = \{p_1, p_2, \ldots, p_n\}$ is a set of analytic processes. $\boldsymbol{p} \in \boldsymbol{P}$ is a tuple $(\boldsymbol{F_p}, \boldsymbol{Trans})$, where $\boldsymbol{F_p} \in \boldsymbol{F}$ is a set of analytic functions and $\boldsymbol{Trans}$ is a set of transition relations between element of $\boldsymbol{F_p}$.
- $\boldsymbol{R} = \{r_1, r_2, \ldots, r_n\}$ is a set of results. Each result $\boldsymbol{r}$ is define as a tuple $(\boldsymbol{i}, \boldsymbol{o}, \boldsymbol{f}, \boldsymbol{p})$, where $\boldsymbol{i}$ is a set of inputs, $\boldsymbol{o}$ is a set of outputs, $\boldsymbol{f} \in \boldsymbol{F}$ is a analytic function produces the result, which can be ***null*** if the result is produced by a analytic process, and $\boldsymbol{p} \in \boldsymbol{P}$ is a analytic process produces the result, which can be ***null*** if the result is produced by a analytic function.

From above definition we can see that a region profile contains all the information that an analyst needs to reproduce an analytic results. It provides a generic description, which is not platform dependent, data dependent, and scenario dependent. Therefore, the analytic results can be easily shared between analysts to reproduce or validate.

The region profile is described in extensible markup Language (XML). Any analytic system that has parser to parse the XML description and have access to the analytic functions will be able to reproduce the results. A typical region profile's structure is listed as follows:

```
<Profile>
    <Name></Name>
    <StartTime></StartTime>
    <EndTime></EndTime>
    <Data>
        <Type></Type>
        <Source></Source>
        <Filter></Filer>
        <Aggregation></Aggregation>
    </Data>
    <Polygons>
        <Polygon>
            <Coordinates></<Coordinates>
        </Polygon>
        …
    </Polygons>
    <Functions>
        <Function>/<Function>
        …
    </Functions>
    <Processes>
        <Process>/<Process>
        …
    </Processes>
    <Results>
        <Result></ Result >
        …
    </Results>
</Profile>
```

In the following section, we will discuss how a region profile is constructed and how it can enhance geo-spatial data analytics through our analytic search system.

4 Region Profile Analytic Search

The region profile analytic search is carried out through two processes in our system: the region profile construction process and the region profile based search process, see Fig. 1. Each process involves several components to help analysts to construct region profiles to capture their interests and use these profiles to find data that match their interest.

4.1 Region Profile Construction Process

The aim of this process is to create a region profile to capture the analysts' interests. This is achieved through few steps:

1. **Highlight a geographical region on map**. At this step, users need to highlight a region on map that is in their interests. This can be done by either loading a predefined shape on map or manually drawing a shape.
2. **Select data to plot in the highlighted region**. At this step, users need to load their data into this region. The data can be from different sources, such as database,

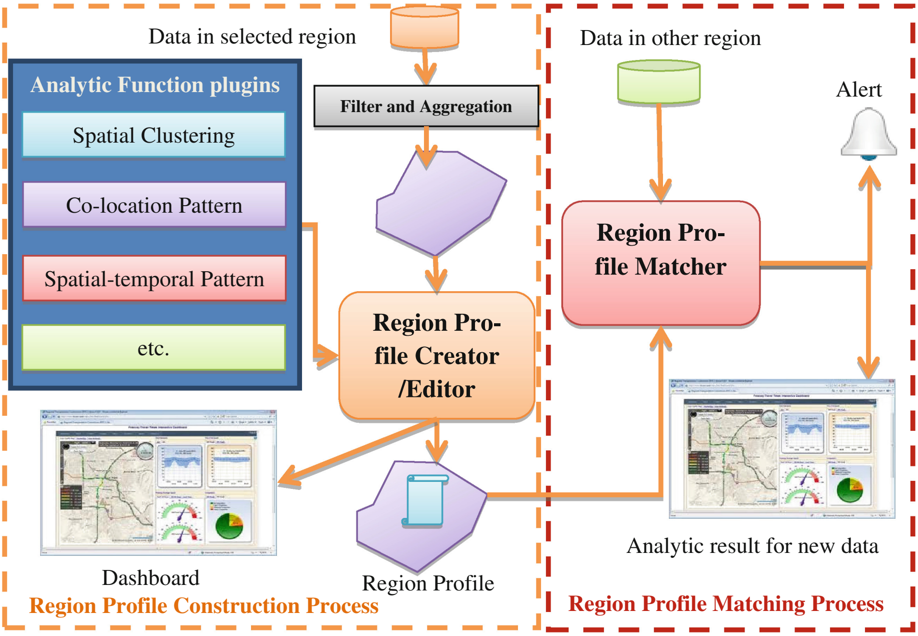


Fig. 1. System structure

Big Data, and spreadsheet etc. The key challenge here is how to bring all the data together and handle them in a uniformed way. In our system, this is done through data pre-processing and integration. Data pre-processing is to make sure that data from different sources are pre-processed into the same format before plotting on map and passing to analytic functions. Data integration is to bring non-geographical data onto the map by linking them with data that have geo-location information. To support the analysis of large amounts of data, users can apply multiple filters on their data and aggregate data in different aspects, including locations, periodicity, and attributes.

3. **Exploratory geo-data analyses with analytic functions and processes**. Once data are loaded into the selected region, users can start to analyses them and decide what should be included in their region profile. The data analysis can start with exploratory analysis if users do not know what the data can tell them. However, the exploratory analysis here is only performed on the subset of user's data within the selected region, not the whole data set. If users know what exactly they are looking for, they can directly select the analytic functions to produce the results. The system provides a selection of analytic functions, such as spatial clustering, Co-location pattern analysis, spatial-temporal pattern analysis etc. External analytic functions are supported through plug-ins. Data can also be analysed by a sequence of analytic functions in order to produce desired results, i.e. analytic processes. The system provides users with a dashboard facility to visualise the analytic results if it is possible, see Fig. 2. (For data protection purpose, the quality of the image is reduced.)

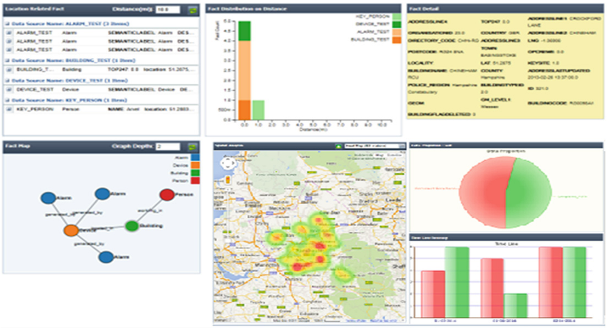

Location Related Fact
Distance(m)
Data Source Name: ALARM_TEST (3 Items)
Data Source Name: BUILDING_TEST (1 Item)
Data Source Name: DEVICE_TEST (1 Item)
Data Source Name: KEY_PERSON (1 Item)
Fact Distribution on Distance
KEY_PERSON
DEVICE_TEST
ALARM_TEST
BUILDING_TEST
Fact Detail
Fact Map
Graph Depth
Alarm
Device
Building
Person
generated_by
working_in
Time Line

Fig. 2. Dashboard view of analytic results

4. **Save desired analytic results as region profile**. Once users decide which analytic functions and processes with their result should be included in a region profile, generate the region profile and save it is very straight forward. The analytic search system will automatically parse the information related to the region profile, e.g. data, analytic functions, processes, related parameters, and their results etc., and generate an XML description. The description can be either saved as a file or stored in a database.

4.2 Region Profile Matching Process

The aim of this process is to help users to find the data that can produce similar results as defined in region profiles. The process itself is very straight forward. Users open a predefined region profile and decide whether let the system automatically find the data that match the profile for them or manually find the data themselves. If the manual option is chosen, users need to drag the shape defined in the profile to other regions of the map and see how the data in those regions match to the region profile. However, there are few challenges in this process need to be discussed.

The first challenge is, in the automatic mode, how the system can efficiently and effectively go through the map space and find the data that satisfy users' region profiles. The simplest way for a system to go through a map space is by dividing the map into portions the same size of the minimum bounding box of the shape in a region profile, see Fig. 3, and going through them one by one. The major problems of this solution are, firstly, an interesting data cluster may be divided into several boxes and therefore lost their analytical features; secondly, this solution is not efficient. As we mentioned before, due to spatial heterogeneity, data are not evenly distributed over a map space. It is totally wasting time and computational power when the system goes through the areas like oceans, desert, and rural areas, which do not contain data at all. To solve this problem, we propose a solution called density based traversal. The algorithm starts with a density analysis in order to know how the data are distributed over a map space and where are the densest points. It will then pick few points with the highest density to start with. At each high density point, the algorithm will move the minimum bounding box of the shape step by step in a hub and spoke style from high density area towards the low density area in all directions that have data, see Fig. 4. At each step, the system will examine how closely the local data are matched to the criteria defined in the region profile. The movement of the minimum bounding box stops when the density level drops below a threshold and the data that have been analysed will be removed from the map space. The whole process will repeat again and again until there is no more data left in the map space or the density of data is lower than a threshold. The distance of each step is configurable. It can be a fixed value or a gradient function that generates values according to the current density, e.g. move slower when the density is high and faster when the density is low. The advantage of this algorithm is that it only analyses the areas in a map space that are relevant and always starts with the most relevant areas. The density based traversal algorithm is performed behind the scenes. Although we talked about map space, there is no need for a map to be present physically. All calculations are done virtually.

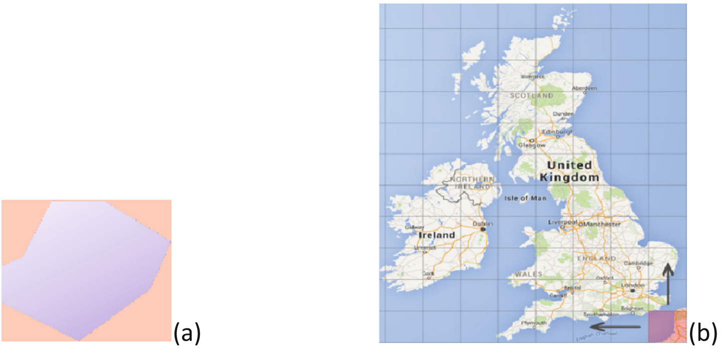


Fig. 3. (a) The Minimum bounding box for a shape (the pink rectangle); (b) Dividing map into minimum bounding boxes (Color figure online)

The second challenge is, how to compare the results generated by a sub-set of data to the region profile in order to know whether they are similar or not. As we discussed previously, a region profile can contain analytic results from a list of analytic functions and processes. If we consider each analytic function or analytic process as an attribute of the region profile, then we can consider a region profile as a complex object with many attributes, where the values of attribute are the results of analytic functions or processes. To compare the similarity between complex objects, we adopted the Term Weighted Cosine Coefficient [2] from the vector based similarity measurement methods to calculate the similarity. Suppose we have two objects O_1 and O_2 and their property vectors $V_1 = (t_1, t_2, \ldots, t_j)$ and $V_2 = (t_1, t_2, \ldots, t_j)$ then we can apply the following formula:

$$Cos\theta_{V_1,V_2} = \frac{\sum_{k=1}^{j} w_{1k} \bullet w_{2k}}{\sqrt{\sum_{k=1}^{j} w_{1k}^2} \bullet \sqrt{\sum_{k=1}^{j} w_{2k}^2}} \quad (1)$$

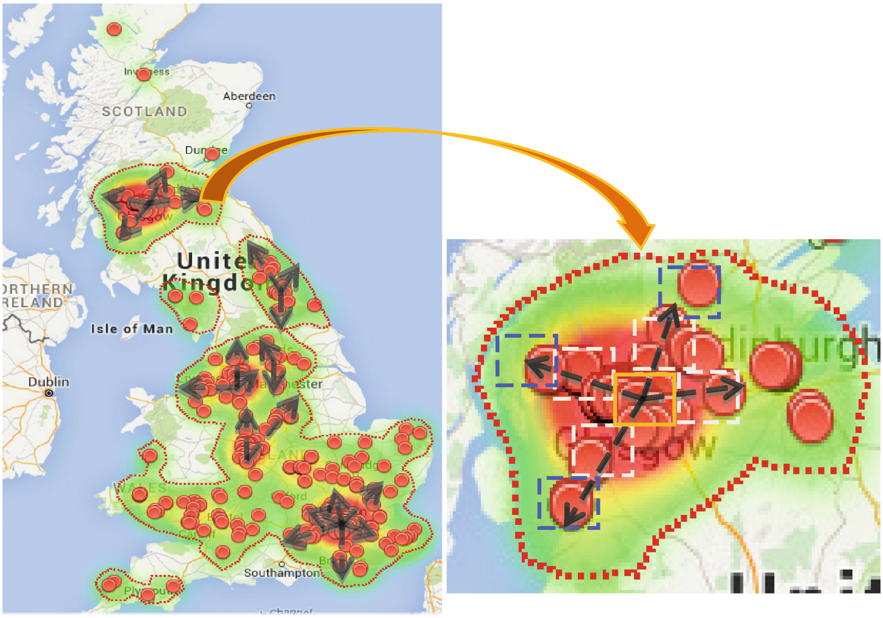


Fig. 4. An illustration of density based traversal

Where, the variable w_{ik} represents the vector V_i's *k-th* term's weighted value. It is normally calculated based on the importance of the properties of an object. Here we assume that although all the results in a region profile are in the analysts' interest, some of the result might have higher impact than others, hence we use weighted value rather than the original value.

At the end of the process, any sub-sets of data that closely matched to the region profile will be reported back to users.

5 Conclusion

As we discussed previously, most of the geo-data analytic tools are exploratory based and lack of facilities that can help analysts to define what they are looking for and quickly find them in their data. In this paper, we introduced the concept of region profile and how it can be used to help geo-data analysts to capture their analytic interests and efficiently find or validate those interests in other data sets.

The system proposed in this paper has a good user base in our organisation. Several patents have been filed around the ideas in this work. In the future work, we will make the system more efficient and scalable by utilising the technologies from the Big Data technology stack, such as Apache Spark.

References

1. ArcGIS. http://www.esriuk.com/software/arcgis. Accessed 20 July 2016
2. Berry, M.W., Drmac, Z., Jessup, E.R.: Matrices, vector spaces, and information retrieval. SIAM Rev. **41**(2), 335–362 (1999)
3. Jacquez, G.M., Greiling, D., Kaufmann, A.: Spatial pattern recognition in the environmental and health sciences: a perspective. IJERPH **7**(4), 1302–1329 (2010)
4. Geovista studio. http://www.geovistastudio.psu.edu. Accessed 20 July 2016
5. Knegt, De, Coughenour, M.B., Skidmore, A.K., Heitkönig, I.M.A., Knox, N.M., Slotow, R., Prins, H.H.T.: Spatial autocorrelation and the scaling of species–environment relationships. Ecology **91**, 2455–2465 (2010)
6. Longley, P.A., Goodchild, M.F., Maguire, D.J., Rhind, D.W.: Geographic Information Science and Systems, 4th edn. Wiley, London (2015). ISBN EHEP003247
7. QGIS. http://www.qgis.org/en/site. Accessed 20 July 2016
8. Chen, S., Yuan, X., Wang, Z., Guo, C., Liang, J., Wang, Z., Zhang, X., Zhang, J.: Interactive visual discovering of movement patterns from sparsely sampled geo-tagged social media data. IEEE Trans. Vis. Comput. Graph. **22**(1), 270–279 (2016)
9. Shestopalov, I.A., Chen, J.K.: Spatiotemporal control of embryonic gene expression using caged morpholinos. Methods Cell Biol. **104**, 151–172 (2011)
10. Wang, J.F., Zhang, T.L., Fu, B.J.: A measure of spatial stratified heterogeneity. Ecol. Ind. **67**, 250–256 (2016)

Segmentation and Enhancement of Low Quality Fingerprint Images

Hasan Fleyeh(✉)

Computer Engineering Department, School of Technology and Business Studies,
Dalarna University, Falun, Sweden
hfl@du.se

Abstract. This paper presents a new approach to segment low quality fingerprint images which are collected by low quality fingerprint scanners. Images collected using such readers are easy to collect but difficult to segment. The proposed approach focuses on automatically segment and enhance these fingerprint images to reduce the detection of false minutiae and hence improve the recognition rate.

There are four major contributions of this paper. Firstly, segmentation of fingerprint images is achieved via morphological filters to find the largest object in the image which is the foreground of the fingerprint. Secondly, specially designed adaptive thresholding algorithm to deal with fingerprint images. The algorithm tries to fit a curve between the gray levels of the pixels of each row or column in the fingerprint image. The curve represents the binarization threshold of each pixel in the corresponding row or column. Thirdly, noise reduction and ridge enhancement is achieved by invoking a rotational invariant anisotropic diffusion filter. Finally, an adaptive thinning algorithm which is immune against spurs is invoked to generate the recognition ready fingerprint image.

Segmentation of 100 images from databases FVC2002 and FVC2004 was performed and the experiments showed that 96 % of images under test are correctly segmented.

1 Introduction

Fingerprint recognition becomes more popular to identify and verify people. There are many commercial applications which require the fingerprint of a person to be read by a sensor in order to be recognized by this application. Recognition of fingerprints often takes place by finding a number of minutiae on the fingerprint image, then the recognition algorithm matches these minutiae with those in application database. Segmentation is the step prior to minute extraction in which a fingerprint image is usually divided into non-overlapping regions; the *foreground* and the *background*. The *foreground* is the area of scanner surface which is in contact with the finger surface and includes the necessary information needed for fingerprint recognition. While the *background* is the noisy area which should be removed by the segmentation process. Therefore, the result of fingerprint segmentation is a fingerprint image in which the background is removed (Akram et al. 2008; Maio et al. 2009). Incorrect segmentation can lead to serious consequences on the recognition as the foreground can be labelled

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 371–384, 2016.
DOI: 10.1007/978-3-319-48743-4_30

as background and vice versa. Furthermore, false minutiae can be generated which gives negative impact on the recognition rate.

When dealing with segmentation fingerprint images, there are two issues to think about. The first one is that the number of images the segmentation algorithm to deal with is huge. A robust segmentation algorithm should be able to deal with all kind of fingerprint images. The second one is that each fingerprint image represents unstructured data which requires specific algorithms to find patterns in the image. Furthermore, dealing with low quality fingerprint images makes extracting these patterns even more difficult.

Most of the segmentation algorithms aim to use one level of features to achieve segmentation (Akram et al. 2008; Bazen and Gerez 2001; Feng et al. 2009; Helfroush and Mohammadpour 2008; Weixin et al. 2009; Yu et al. 2008). In general, there are two approaches for fingerprint segmentation: block-wise based or pixel-wise based (Ren et al. 2008). In the block-wise approach, the fingerprint image is divided into blocks and each block is classified into foreground or background based on features calculated for the block. While in the pixel-wise method, segmentation is achieved on the pixel level.

This paper proposes a new technique to segment fingerprint images. Segmentation is achieved by morphological filters to find the largest object in the fingerprint image. An adaptive threshold algorithm specially designed for fingerprint images is also proposed. This algorithm considers the gray levels of each row or column in the fingerprint image as 2D function and fits a curve among the highest and lowest values. The fitted curve represents the adaptive threshold values for each pixel in this row or column. The thresholded fingerprint image is enhanced by removing noise and reconnecting separated ridges by rotational invariant anisotropic diffusion filter. Finally an adaptive thinning algorithm which is able to remove all spurs is invoked to generate the final fingerprint image.

The rest of the paper is organized as follows. In the next section state of the art of fingerprint segmentation is presented. In Sect. 3 the proposed method is illustrated. The experimental results based on the proposed method are given in Sect. 4, and in Sect. 5 the conclusion is presented.

2 Literature Review

In recent years, biometric authentication became a big field of research because of its importance to identify and verify people. Automated Fingerprint Identification System (AFIS) is one among many other fields which was developed rapidly. In this section, the recent state of the art is presented.

Sankaran et al. ((2017) proposed a method to automatically segment latent fingerprints in order to distinguish between ridge and non-ridge patterns. The authors invoked machine learning algorithms to achieve the latent segmentation. The fingerprint image was divided into non-overlapping blocks and for each block a number of features were computed and fused to construct a feature vector. Features include Saliency, Intensity, Gradient Ridge, and quality features were fused to create the feature vector. A random forest classifier was employed to segment the image into foreground and background blocks. Results showed high segmentation accuracy of about 96 % on inked fingerprint datasets.

Thai et al. (2016) proposed a novel factorized directional bandpass (FDB) segmentation method for texture extraction based on the directional Hilbert transform of a Butterworth bandpass Directional Hilbert Butterworth

Bandpass filter (DHBB) filter interwoven with soft-thresholding. The original image was transformed into the Fourier domain and filtered first by the DHBB factor obtaining 16 directional sub-bands. Next, soft-thresholding was applied to remove spurious patterns. The feature image was reconstructed from these sub-bands using a second DHBB factor. Finally, the feature image was binarized and the ROI is obtained by morphological operations.

Thai and Gottschlich (2016) developed a segmentation method by global three-part decomposition (G3PD). Based on global variational analysis, the G3PD method decomposed the fingerprint image into cartoon, texture and noise parts. After decomposition, the foreground region was obtained from the non-zero coefficients. The proposed method was evaluated by the segmentation of 10560 images.

Ezeobiejesi and Bhanu (2016) proposed an algorithm to segment latent fingerprint which was based on fractal dimension features and weighted extreme learning machine. The feature vectors, which were built from the local fractal dimension features, were invoked as input to a weighted extreme learning machine ensemble classifier. The result of classification was two classes; fingerprint and non-fingerprint classes. The proposed segmentation algorithm was evaluated by achieving better results than the state of the art regarding the false detection rate (FDR) and overall segmentation accuracy compared to the existing approaches.

Nimkar and Mishra (2015) developed an algorithm for fingerprint segmentation. The proposed algorithm was named as Adaptive (scale) and Orientation (vector). The basic idea of the proposed algorithm was originated from the total variation models, along with two features of fingerprints; namely, scale and vector. The result of the algorithm was to decompose the fingerprint image into two regions; noisy and texture. The algorithm was tested on two different fingerprint datasets and PNSR was invoked to check the efficiency of the algorithm.

Carneiro et al. (2014) achieved a comparative study to analysis four thresholding techniques (Niblack, Bernsen, Fisher, Fuzzy), two thinning techniques (Stentiford and Holt) and a feature extraction (Cross Number) technique for fingerprint applications. The authors tested and analyzed the algorithms on a set of 160 fingerprint images. The results pointed out the positive and negative sides of the different algorithms.

3 The Proposed Approach

The proposed approach for fingerprint segmentation and enhancement is depicted in Fig. 1. It consists of five stages:

1. **Pre-processing**

Fingerprint images with low contrast, false traces ridges or noisy complex background cannot be segmented correctly. Therefore, such images should be enhanced. The pre-processing applied in this paper is mean and variance normalization. A fingerprint image $\mathbf{I}[x, y]$ is normalized by specifying its desired mean and variance values denoted m_0 and v_0 as shown in Eq. 1.

$$\mathbf{I}'[x, y] = \begin{cases} m_0 + \sqrt{(\mathbf{I}[x, y] - m)^2 . v_0 / v} & \textit{if } \mathbf{I}[x, y] > m \\ m_0 - \sqrt{(\mathbf{I}[x, y] - m)^2 . v_0 / v} & \textit{otherwise} \end{cases} \tag{1}$$

This process produces a fingerprint image $\mathbf{I}'[x, y]$ according to m_0 and v_0. In this equation, m and v are the mean and variance of the fingerprint image $\mathbf{I}[x, y]$.

2. Segmentation

The proposed segmentation approach segments the fingerprint image in two non-overlapping regions; the foreground and the background according to the following relationship:

$$\mathbf{I} = \cup (Fo, Ba) \tag{2}$$

where **I** is the fingerprint image, *Fo* is the foreground region and *Ba* is the background region.

Since the fingerprint foreground represents the object with the largest area in the image, locating this object in the image means locating its foreground. Figure 2 depicts the steps followed to isolate this object from the rest of the image. The process starts by binarizing the fingerprint image using the Otsu thresholding method (Otsu 1979). To isolate the foreground from the rest of the image, dilation is applied to force the detached ridges to attach to each other. A square structuring element whose size is related to that of the image under consideration is created. The fingerprint foreground becomes a large single object dominating the image. By applying a modified version of connected component labelling algorithm (Suzuki et al. 2003) and targeting the largest object, the foreground of the fingerprint is located and extracted. Extraction of this object takes place by a simple IF THEN rule which checks the presence of a white pixel in the image contacting the largest foreground object. Figure 4C depicts the largest object in the fingerprint image while Fig. 4D presents the results of this segmentation.

3. Adaptive Thresholding

Due to the fact that gray levels of pixels representing fingerprint's ridges varies widely from an image to another and from one position in a certain image to another one, traditional thresholding algorithms may not produce robust results. Therefore, the need for an adaptive algorithm which eliminates such problems is essential. In this paper, a special thresholding algorithm for fingerprints analysis is proposed. Consider a strip of one pixel width taken laterally or longitudinally anywhere in a fingerprint image where ridges and valleys exist. Plotting this strip gives a wave similar to that shown in Fig. 3. In this plot, the x-axis represents the location of the pixels in the strip, while the y-axis is the gray level of each pixel. The highest amplitude point of each cycle of the wave represents a ridge while the lowest amplitude point corresponds to a valley.

The main idea of the proposed adaptive thresholding algorithm is find a curve which fits the ridge-valley wave and separates it into two parts; an upper part

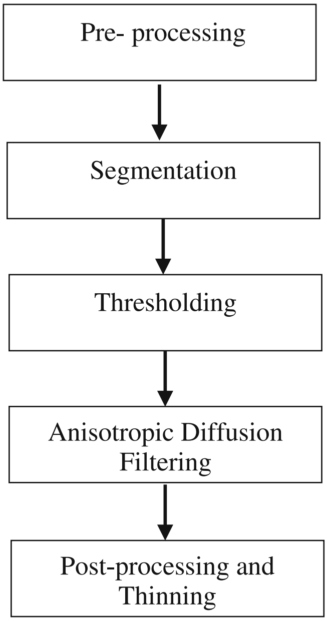


Fig. 1. The proposed Approach for fingerprint segmentation and enhancement.

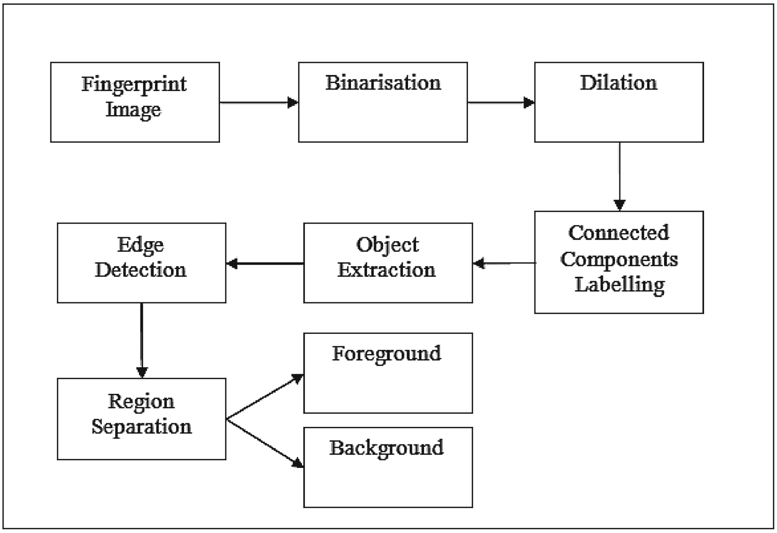


Fig. 2. Segmentation of fingerprint images.

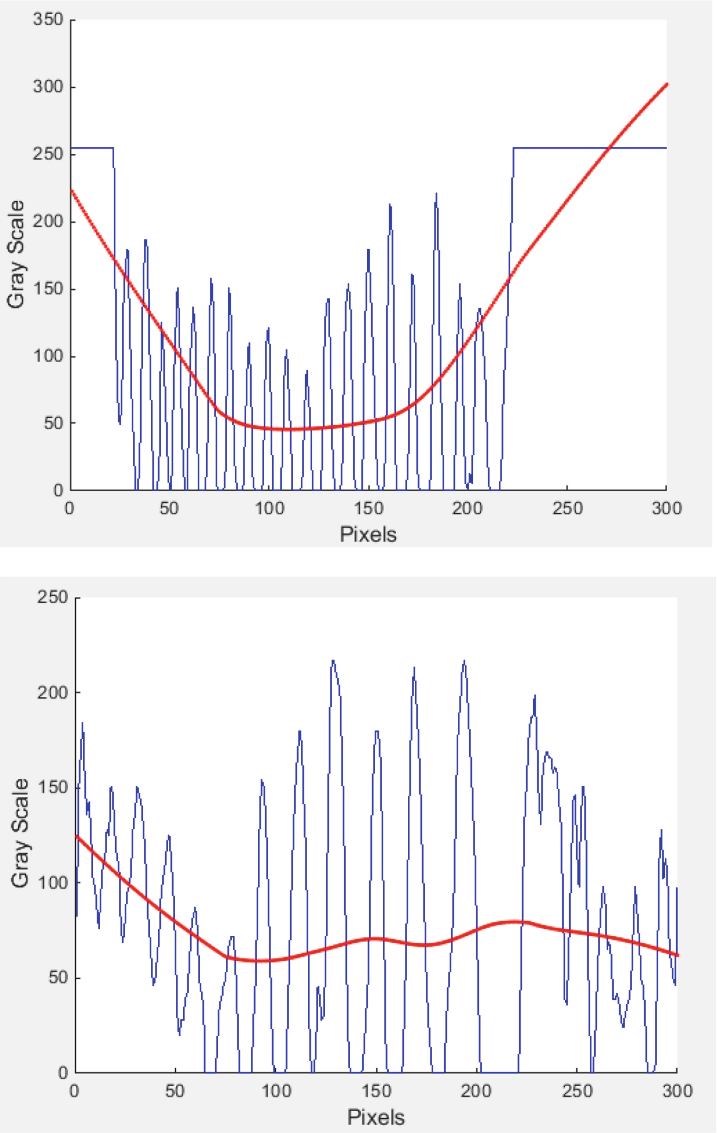


Fig. 3. Applying LOWESS fitting on horizontal strips (above) and vertical one (below). The blue colour is the ridge-valley system of one strip and the red curve is the resultant fitting curve. (Color figure online)

represents the ridges and the lower one represents the valleys. The fitting algorithm which can achieve this job is called Locally Weighted Scatter-plot Smoothing (LOWESS) (Cleveland 1979; Cleveland and Devlin 1988). This is a method to fit a smooth curve between two variables where one variable is the ridge and the other one is the valley. The method is nonparametric because the linearity assumptions of

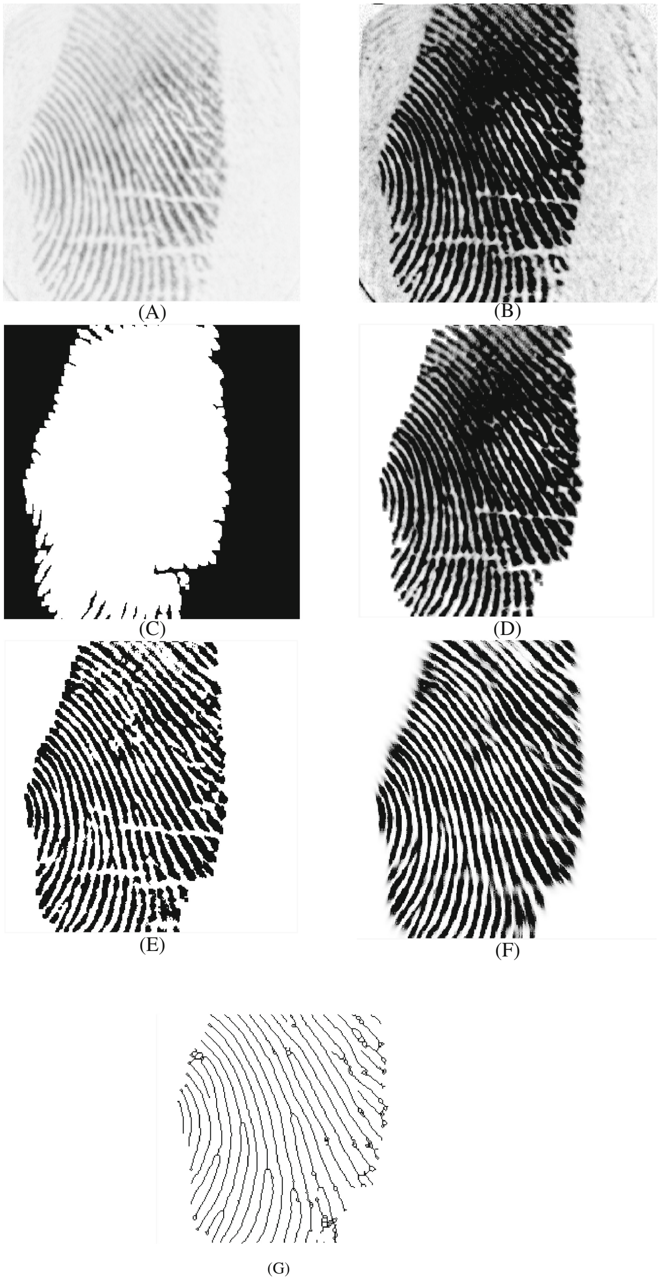


Fig. 4. Details of the proposed approach. (A): Original Image. (B): Fingerprint image after Pre-processing. (C): Image of the largest object. (D): Extracted foreground. (E): Thresholding a fingerprint strip by LOWESS. (F): Rotational Invariant Anisotropic Diffusion Filter. (G): Thinning and Post-processing

conventional regression methods are relaxed. Therefore, the overall uncertainty is measured as how well the estimated curve fits the population curve. Applying this method upon the ridge-valley curve depicted in Fig. 5 generates a curve (the red curve) which fits between the ridges and the valleys. The intersection of this curve with the ridge–valley system is a unique value of threshold calculated for each cycle of the wave (a ridge and a valley). Once these adaptive threshold values are computed, the strip is converted into black and white. This operation is applied for each image row from the top to the bottom and each column from the left to the right. The final binary image is generated by an OR operation of the two images generated from the horizontal and vertical strips which is depicted in Fig. 4E.

4. Anisotropic Diffusion Filtering

Many fingerprint matching systems employ minutiae for matching. With the presence of noise in the image, many true minutiae can be missed and false minutiae can be detected instead. Therefore, the recognition process will be affected. In order to avoid these errors, it is essential to improve the fingerprint image quality. Anisotropic Diffusion filter is designed to reduce the noise in images while preserving the region edges, and to smooth along the image edges removing gaps due to noise.

The basis of the method was introduced by Weickert (Weickert and Scharr 2002). Modifications and improvements were presented by (Kroon and Slump 2009) (Gottschlich and Schönlieb 2012; Kroon et al. 2010).

The method consists of two steps. In the first step, the image is described by a structure tensor called the second-moment matrix. While in the second step, the structure tensor is transformed into a diffusion tensor for edge enhancing diffusion filtering.

The details of this approach is given in (Kroon and Slump 2009) which can be summarized as follows:

1. Smooth the image by a Gaussian filter.
2. Calculate Hessian from every pixel of the Gaussian smoothed image.
3. Gaussian Smooth the Hessian.
4. Calculate eigenvectors and eigenvalues of the image from step 3. Note that image edges give large eigenvalues, and the eigenvectors corresponding to those large eigenvalues describe the direction of the edge.
5. The eigenvectors are used as diffusion tensor directions.
6. The diffusion is performed by a finite difference scheme.
7. Back to step 2, till a certain diffusion time is reached.

Applying this approach on the fingerprint images produced by the former step will improve image quality by removing the noise and enhancing and preserving ridge's edges. Results of this enhancement is depicted in Fig. 4F.

5. Post-processing

In this stage, the skeleton of the enhanced fingerprint image is generated. The skeleton algorithm invoked in this stage was designed to handle situations where spurs are to be minimized which fits the requirements of fingerprint recognition applications. It is assumed that skeleton points are those which sit at the center of a circle that touches the edge of the shape to be skeletonized at multiple points. The gray-level of each pixel on the skeleton depends upon the shortest distance to travel around the perimeter of the shape to be skeletonized to connect the most distant two points. Thus spurs in the skeleton due to small edge perturbations will have low intensity even if they are very long which fits the requirements of fingerprint applications. Finally a threshold should be selected depending on the expected size of any noisy protrusions in the silhouette (Howe 2016). An example of an image which is treated with post-processing is depicted in Fig. 4G.

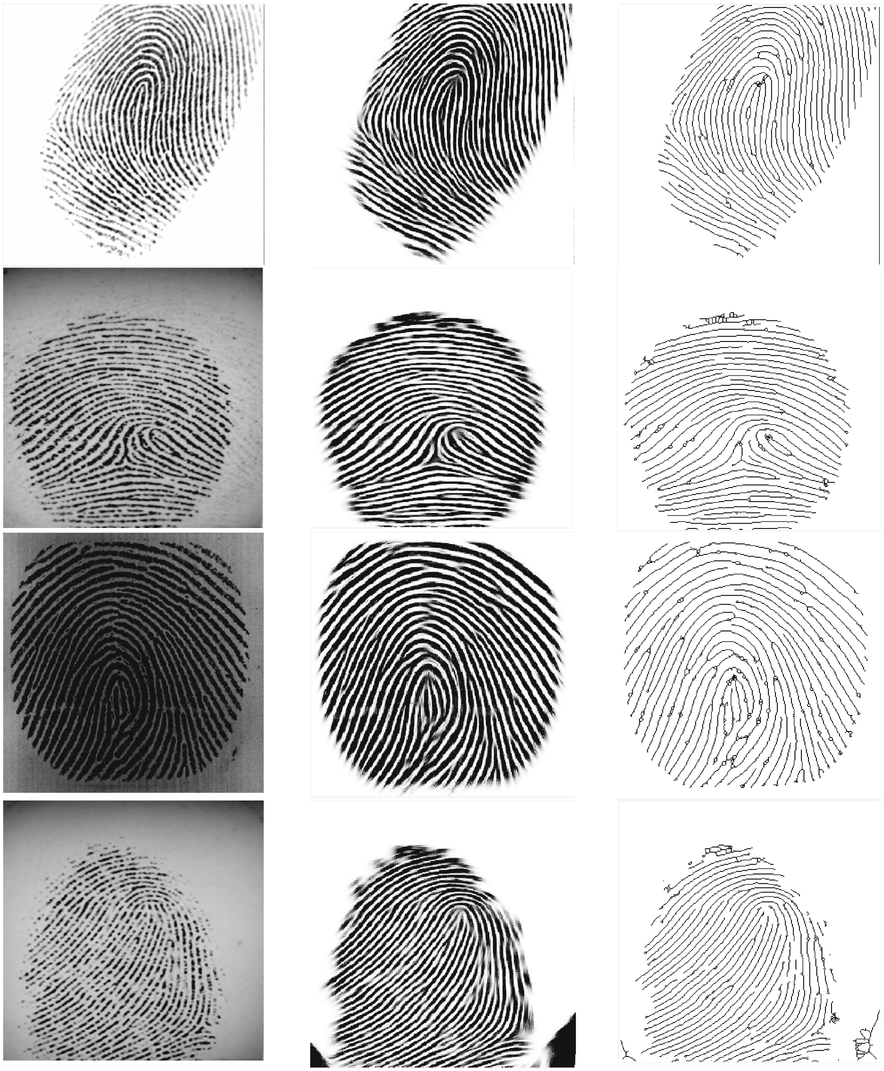


Fig. 5. Segmentation and enhancement results from FVC2002. Left column: Original image. Middle column: Results of rotational invariant anisotropic diffusion filter. Right column: Thinning and post processing. Row 1: FVC2002-DB1-A-9_3. Row 2: FVC2002-DB2-A-2_7. Row 3: FVC2002-DB3-A-4_3. Row 4: FVC2002-DB4-A-17_1

4 Experiments and Results

To test the proposed algorithm, two different datasets are employed. The first one is the second fingerprint verification competition FVC2002 (2002) which consists of 4 datasets (3 real and 1 synthetic). There are 31 participants who voluntarily submit their fingerprints. The second dataset is FVC2004 (2004) which has fingerprints of 43

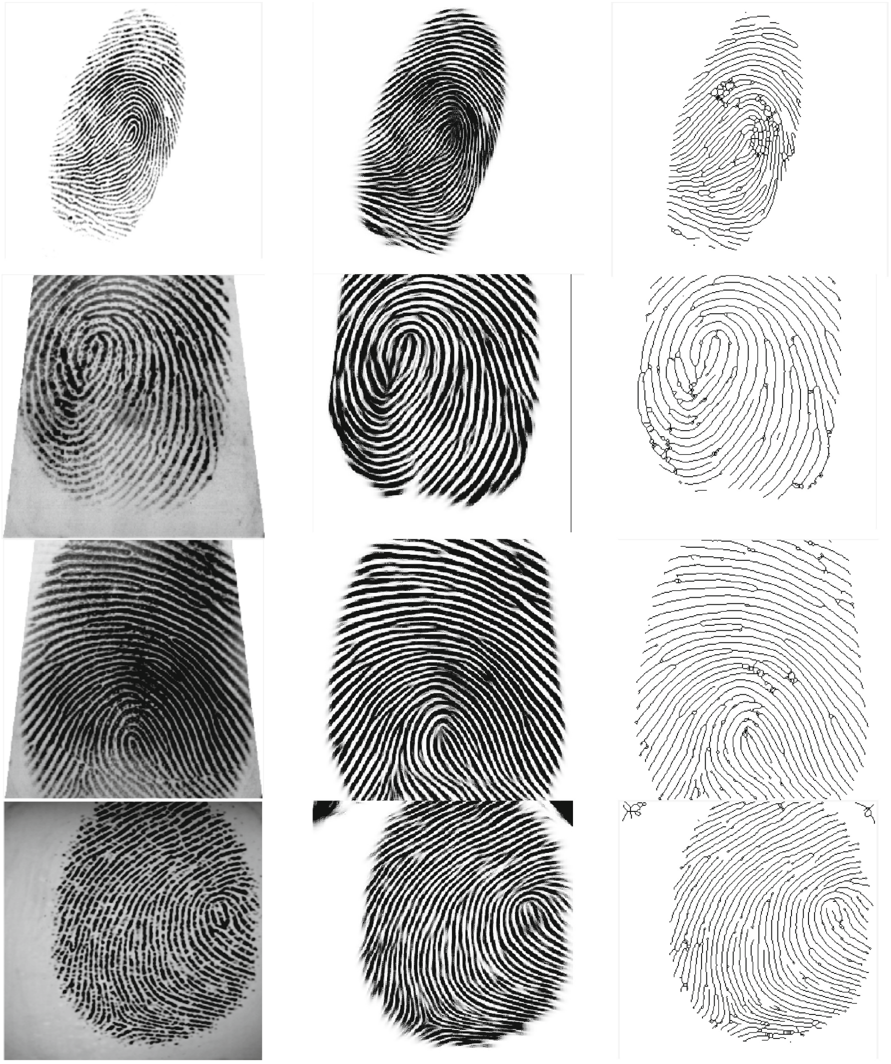
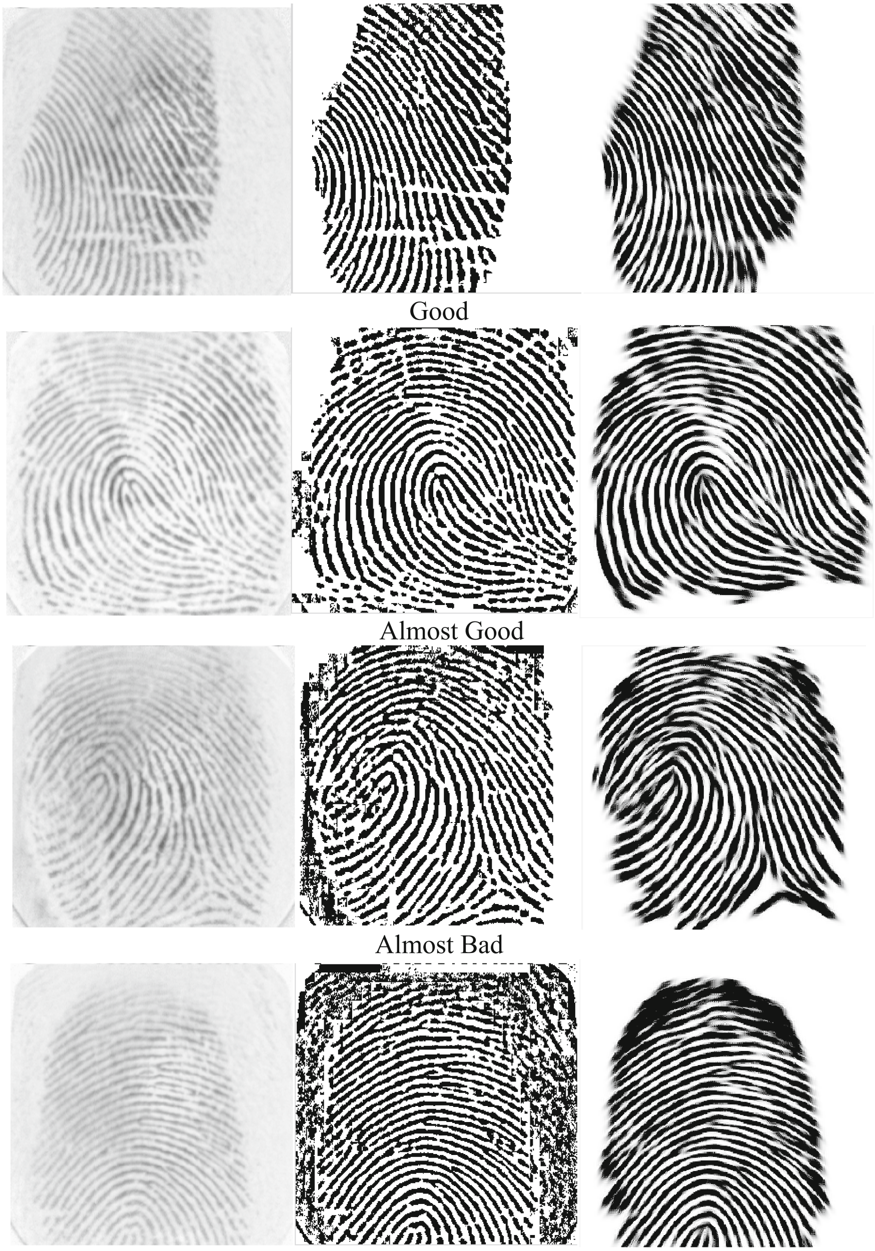


Fig. 6. Segmentation and enhancement results from FVC2004. Left column: Original image. Middle column: Results of rotational invariant anisotropic diffusion filter. Right column: Thinning and post processing. Row 1: FVC2004-DB1-B-107_6. Row 2: FVC2004-DB2-B-103_1. Row 3: FVC2004-DB2-B-109_4. Row 4: FVC2004-DB2-B-105_4



Good

Almost Good

Almost Bad

Bad

Fig. 7. A comparison with the algorithm described by (Fleyeh and Jomaa 2010). Left column: Original images. Middle column: segmentation from (Fleyeh and Jomaa 2010). Right column: results from current approach.

participants including 29 industrial, 6 academics and 8 independent developers. It is very important to mention that different scanning sensors were used to collect the fingerprint images and dealing with FVC2004 databases is much harder than 2002 due to the perturbations deliberately introduced.

The proposed approach was tested by 100 fingerprint images which were selected randomly and without repetition from the FVC2002 database DB1-A, DB2-A, DB3-A, and DB4-A, and from FVC2004 DB1-B, DB2-B and DB4-B. These images represent unstructured data which should be cleaned and prepared for classification. Wrong segmentation may increase false minutiae in the image and hence reduce classification rate. On the hand, undetected true minutiae has a negative impact on the classification rate. The validation test showed that the proposed algorithm could segment 96 % of the images under test. Samples of segmented images from datasets FVC2002 and FVC2004 are depicted in Figs. 5 and 6, respectively.

A comparison with the method proposed by Fleyeh and Jomaa (2010) to segment images from FVC2000 shows that the current approach performs much better than the other one. The results depicted in Fig. 7 show that segmentation is good even with images which are previously classified as bad by the other approach. It can be seen that great improvements have been achieved by the current approach compared to the former one. These improvements are not only clear with images formerly classified as bad or almost bad, but also for images classified as good and almost good.

5 Conclusion

The problem of fingerprint segmentation is one of the pattern classification paradigms which are not fully solved yet. Fingerprint images collected as forensic evidences suffer from background noise. This paper proposes a new approach to segment and enhance low quality fingerprint images. Segmentation was achieved by morphological operations. An adaptive thresholding approach which was specially designed for fingerprint images was proposed and tested. A rotation invariant anisotropic diffusion filter was invoked to remove noise and enhance and preserve ridges. This step enhances ridge structure and reduces false minutiae in the segmented images. Finally, an adaptive thinning algorithm which was able to remove spurs was included in the proposed approach.

A set of experiments were performed to evaluate the proposed approach which showed high robustness. The proposed approach was able to segment 96 % of images used for testing. All images which were segmented as almost good, almost bad and bad can now be segmented as good, which indicates that high robustness is achieved.

References

Akram, M.U., Nasir, S., Tariq, A., Zafar, I., Khan, W.S.: Improved fingerprint image segmentation using new modified gradient based technique. Paper presented at the 2008 Canadian Conference on Electrical and Computer Engineering, Niagara Falls, Canada, 4–7 May 2008

Bazen, A.M., Gerez, S.H.: Segmentation of fingerprint images. Paper presented at the ProRISC 2001 Workshop on Circuits, Systems and Signal Processing, Veldhoven, The Netherlands, November 2001

Carneiro, R., Bessa, J., Moraes, J.D., Neto, E., Alexandria, A.D.: Techniques of binarization, thinning and feature extraction applied to a fingerprint system. Int. J. Comput. Appl. **103**(10), 1–8 (2014)

Cleveland, W.: Robust locally weighted regression and smoothing scatterplots. J. Am. Stat. Assoc. **74**, 829–836 (1979)

Cleveland, W., Devlin, S.: Locally weighted regression: an approach to regression analysis by local fitting. J. Am. Stat. Assoc. **83**, 596–610 (1988)

Ezeobiejesi, J., Bhanu, B.: Latent fingerprint image segmentation using fractal dimension features and weighted extreme learning machine ensemble. Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition Workshops (2016)

Feng, W., Xiuyou, W., Lin, X.: An improved fingerprint segmentation algorithm based on mean and variance. Paper presented at the 2009 International Workshop on Intelligent Systems and Applications, Wuhan, China, 23–24 May 2009

Fleyeh, H., Jomaa, D.: Segmentation of Low Quality Fingerprint Images. Paper presented at the IEEE International Conference on Multimedia Computing and Information Technology (MCIT-2010), Sharja, UAE, 2–4 March 2010

FVC2002 (2002). http://bias.csr.unibo.it/fvc2002/

FVC2004 (2004). http://bias.csr.unibo.it/fvc2004/

Gottschlich, C., Schönlieb, C.: Oriented diffusion filtering for enhancing low-quality fingerprint images. IET Biom. **1**, 105–113 (2012)

Helfroush, M., Mohammadpour, M.: Fingerprint segmentation. Paper presented at the 3rd International Conference on Information and Communication Technologies: From Theory to Applications, Damascus, Syria (2008)

Howe, N.: Implementation of Contour-Pruned Skeletonization (2016). http://cs.smith.edu/~nhowe/research/code/index.html#binarize

Kroon, D., Slump, C.: Coherence filtering to enhance the mandibular canal in cone-beam CT data. Paper presented at the IEEE-EMBS Benelux Chapter Symposium (2009)

Kroon, D., Slump, C., Maal, T.: Optimized anisotropic rotational invariant diffusion scheme on cone-beam CT. Paper presented at the International Conference on Medical Image Computing and Computer-Assisted Intervention (2010)

Maio, D., Maltoni, D., Jain, A., Prabhakar, S.: Handbook of Fingerprint Recognition, 2nd edn. Springer, London (2009)

Nimkar, R., Mishra, A.: Fingerprint segmentation using scale vector algorithm. Paper presented at the IEEE 2015 5th International Conference on Communication Systems and Network Technologies (CSNT) (2015)

Otsu, N.: A threshold selection method from gray level histogram. IEEE Trans. Syst. Man Cybern. SMC **9**(1), 62–66 (1979)

Ren, C., Yin, Y., Ma, J., Yang, G.: A linear hybrid classifier for fingerprint segmentation. Paper presented at the IEEE 4th International Conference on Natural Computation, Jinan, China (2008)

Sankaran, A., Jain, A., Vashist, T., Vatsa, M., Singh, R.: Adaptive latent fingerprint segmentation using feature selection and random decision forest classification. Inf. Fusion **34**, 1–15 (2017)

Suzuki, K., Horiba, I., Sugie, N.: Linear-time connected component labelling based on sequential local operations. Comput. Vis. Image Underst. **89**, 1–23 (2003)

Thai, D., Gottschlich, C.: Global variational method for fingerprint segmentation by three-part decomposition. IET Biom. **5**(2), 120–130 (2016)

Thai, D., Huckemann, S., Gottschlich, C.: Filter design and performance evaluation for fingerprint image segmentation. PLoS ONE **11**(5), 154–160 (2016)

Weickert, J., Scharr, H.: A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance. J. Vis. Commun. Image Represent. **13**, 103–118 (2002)

Weixin, B., Deqin, X., Yi-Wei, Z.: Fingerprint segmentation based on improved active contour. Paper presented at the IEEE Computer Society International Conference on Networking and Digital Society (2009)

Yu, C., Xie, M., Qi, J.: An effective algorithm for low quality fingerprint segmentation. Paper presented at the IEEE 3rd International Conference on Intelligent System and Knowledge Engineering, Chengdu, China (2008)

Feature Selection and Bleach Time Modelling of Paper Pulp Using Tree Based Learners

Karl Hansson, Hasan Fleyeh(✉), and Siril Yella

Computer Science Department, Dalarna University, 791 31 Falun, Sweden
{khs,hfl,sye}@du.se
http://www.du.se

Abstract. Paper manufacturing is energy demanding and improved modelling of the pulp bleach process is the main non-invasive means of reducing energy costs. In this paper, time it takes to bleach paper pulp to desired brightness is examined. The model currently used is analysed and benchmarked against two machine learning models (Random Forest and TreeBoost). Results suggests that the current model can be superseded by the machine learning models and it does not use the optimal compact subset of features. Despite the differences between the machine learning models, a feature ranking correlation has been observed for the new models. One novel, yet unused, feature that both machine learning models found to be important is the concentration of bleach agent.

Keywords: Feature selection · Machine learning · CFS · Random forest · TreeBoost · XGBoost · Paper manufacturing

1 Introduction

Paper manufacturing is intrinsically energy demanding. According to the Confederation of European Paper Industries, its members consumed about 101 TWh in 2013 [1]. To put this in perspective, the EU-28 countries consumed a total of 3262 TWh electricity in the same year, roughly a third were consumed by industries [2]. The paper manufacturing in Europe thus accounted for roughly 8.5 % of the total usage of electrical energy in the industrial sector. The collaborative partner in this work is one of the largest paper mills in Sweden. Even relatively small improvements in terms of energy efficiency leads to substantial energy savings and competitive advantages in terms of cost effectiveness. The time when paper production goes from producing paper of one quality to another is called a changeover. During changeovers, produced paper cannot be sold since the quality of the product is not in any marketable state. Since the production is continuous, the manufacturing plant still consumes the same amount of energy as in a non-changeover state. Paper produced during a changeover must be recycled and there is, therefore, a two folded energy waste. Reducing changeover time is among the most effective mean a paper manufacturer can do to reduce costs. A success factor for reducing the changeover time is to better understand

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 385–396, 2016.
DOI: 10.1007/978-3-319-48743-4_31

and model the bleaching process, a process that is hard to observe directly as it possesses long lead times. If brightness of the paper is to be changed, a bleaching process must be initiated hours in advance. Better prediction of bleach time would improve planing and preparations for coming changeovers. Currently the company models the bleach process by looking at mass transport of paper pulp within the paper production plant.

The main objective of this article is to investigate and ensure whether if all the necessary features are incorporated in the current model for predicting the bleach time. The proposed method is based on a pre-study, that outlines ideas of feature selection and modelling of high dimensional manufacturing systems [3]. The investigation is done by first using Hall's filter technique Correlation-based Feature Selection [4], and append this new feature set to the existing feature set used by the pre-existing model. Collected data from the paper plant was then invoked to benchmark the current model and also to train two strong models based on Brieman's Random Forest [5] and Friedman's TreeBoost algorithm [6]. The models are then benchmarked against each other to examine if the extended feature set offered means of more precise predictions of the bleach time using novel data. The machine learning models are compelling as both offer functionality for assessing feature importance. This article presents and analyse feature rankings suggested by the new models.

This article is organized as follows. In Sect. 2, available data is discussed. Section 3 discusses the models employed in this work. Section 3 presents measuring of variable importance and in Sect. 4 experimental results are shown. Results are discussed in Sect. 5 and the article is concluded in Sect. 6.

2 Data

It is essential to monitor manufacturing plants. Sensors are placed throughout the entire process, with the primary function of letting operators and control loops observe the current state of production. Such sensor readings are stored in a data warehouse to let the manufacturer evaluate past performance of the production. In this case, thousands of different variables have been stored in a data warehouse. In this article, only data stored between the bleach tower and the point where the brightness of the paper pulp is measured are used.

The manufacturer continuously stores a product code describing the type of paper that is currently produced. The product code contains information regarding several properties of the paper, one among them is brightness. By looking at the recorded product codes in the period March 2013 to October 2015, it could be determined at what timestamps the plant started to produce a new product with a different brightness. Furthermore, given these timestamps it was possible to match if changes in the concentration of bleaching agent in the bleach tower had preceded the change in brightness. A total of 231 observations of changes in bleach agent within 12 hours of a brightness change was identified. The time difference between change in bleach agent level and brightness change in the final product was used as the target feature to evaluate model performance. If several

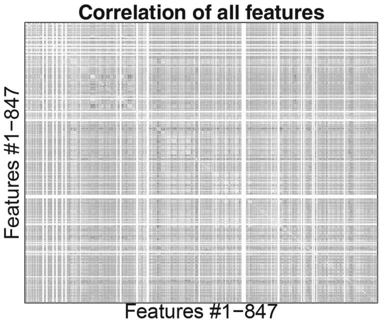


Fig. 1. Correlation between the 847 variables. Lighter greys indicates higher correlation, completely white lines are features with zero variance.

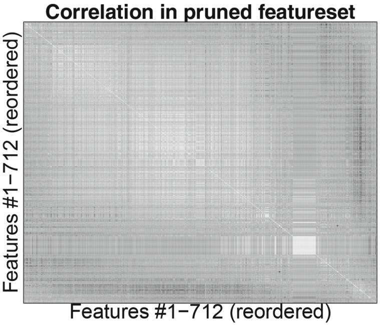


Fig. 2. Correlation between the 712 features after removal of zero variance features. Reordered rows and columns to visualize closely correlated features.

changes in the bleach agent occurs within a time frame, the one closest to the brightness change was the one considered. One problem with the target value worth commenting on is that it included both the time it takes for the bleaching as well as the time of the actual changeover. There are 847 different candidate features extracted from the companies data warehouse to create a dataset which were used to predict bleach time. These features represents set points and sensor readings from the pulp manufacturing part of the plant and the paper machine itself. The vast majority of the features are numerical, but some of them can be treated as categorical as they only take binary values. In industrial environments, missing values are common and often carries information regarding the status of the machinery. It was neither suitable to omit nor impute them. However, 135 of the features never change in any of the observations. Figure 1 depicts all features and their corresponding correlations, lighter greys signifies a higher correlation and the white lines are features with zero variance (never changing). The features with zero variance are removed since they cannot add any information to the data driven models. Removing the non-changing ones reduces the number of features to 712, where 80 of them are binary. Furthermore, a number of features are fully correlated to each other, because of the nature of the production process. This is not directly apparent in Fig. 1 but if the rows and columns of the correlation matrix are reordered the correlation becomes clear as shown in Fig. 2.

3 Models

In this section, three different types of models are described. First, the current model of the bleach time in the paper plant is presented. It is then followed by presenting the two machine learning algorithms that are proposed to be used to identify additional features.

3.1 The Current Model

The current model used by the paper manufacturer to predict bleach time has been constructed from a set of differential equations that describes flow of paper pulp between storage tanks in the paper manufacturing plant. In total 7 features, $\{x_1, x_2, \ldots, x_7\}$, and 3 numerical constants, $\{\alpha_1, \alpha_2, \alpha_3\}$, are used to predict the bleach time given the current state of the production plant. These features and numerical constants have been selected by domain experts. Feature and constant names have been replaced by dummy names due to confidentiality reasons. The model itself is shown in Eq. 1.

$$\hat{y} = \frac{\alpha_1 x_1 x_2}{(\alpha_2 - x_3)(x_4(\alpha_2 - x_6) - \alpha_2 x_5)(\alpha_2 - x_7)} + \alpha_3 \tag{1}$$

3.2 Machine Learning Models

Tree based ensemble learners have in the past few years emerged as the top performing algorithms for modelling complex systems based on structured data [7–10]. The strength of ensembles has been analytically confirmed by Tumer and Ghosh [11]. The authors proved that an ensemble of weak learners, which are not strongly correlated to each other, improves the combined generalization ability. Ensemble techniques are especially useful when the training data is in limited supply. Single decision trees are sensitive to the training data in the sense that minor changes in the training data can produce widely different trees [12,13] which also motivates the use of ensemble techniques. Both machine learning models used in this article are based on the CART tree, introduced by Breiman et al. [12], which has been widely employed due to its simplicity and predictive capabilities. The CART tree is a binary tree that splits the feature space in two parts at each node, leafs in the tree describes the final regression/classification. A CART tree, $h(\mathbf{x})$, partitions the input space into J disjunct regions, $R_1, \ldots, R_J$. A value is associated with each of the regions, b_j, it is this value that constitutes the prediction of individual trees. The individual tree model is expressed in Eq. 2, where $1(\mathbf{x} \in R_{jm})$, is the binary indicator function.

$$h(\mathbf{x}) = \sum_{j=1}^{J} b_j 1(\mathbf{x} \in R_j) \tag{2}$$

3.3 Random Forest

The Random Forest, RF, is a modelling technique that employs bagging [14], which reduces variance and overfitting of a classifier. Bagging means that given a training set $\mathcal{D}$ with size n, re-sample $\mathcal{D}$ uniformly with replacements into m new datasets $\mathcal{D}_i$ each with size n'. Usually $n' = n$ is chosen, then $1 - 1/e$ of the samples are expected to be unique within each of the produced subsets [15]. After re-sampling, a model of choice is fitted for each $\mathcal{D}_i$. The final prediction is made by having a majority vote from the weak learners, each being equally weighted. RF was introduced by Breiman [5], RF extends the idea of bagging by not only sub-sample datasets but also by random sampling of which features each tree in the forest uses. Predictions using an RF model is done via a majority vote as shown in Eq. 3.

$$\hat{y}(\mathbf{x}) = \frac{1}{K}\sum_{k=1}^{K} h_k(\mathbf{x}) \tag{3}$$

Measuring variable importance in an RF works by randomly permuting individual feature to measure how much the sum of squares errors (SSE) increases when a feature becomes noisy. The rationale behind this argument is that when an important feature randomly changes the prediction error increases. Given feature data $\mathcal{D} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_m}\}$, corresponding targets $\mathbf{Y}$ and an RF model, $H(\mathbf{x})$, variable importance is then calculated as follows.

1. Calculate the unaltered SSE, $e_0 = \sum (Y - H(\mathcal{D}))^2$.
2. Permute each feature, $i \in \{1, \ldots, m\}$ while keeping the rest fixed. The dataset $\mathcal{D}_i$ denotes a permuted i:th feature.
3. Recalculate SSE, $e_i = \sum (Y - H(\mathcal{D}_i))^2$ for each $\mathcal{D}_i$.
4. For each i calculate the variable importance ratio, $I_i = e_i/e_0$

3.4 TreeBoost

In 2001 Friedman [6] proposed a gradient boosting machine extending on Schapire's work with boosting [16]. The algorithm is generic with regards to predictor, but can easily be optimized for CART trees, this variant is called TreeBoost. In this article an R implementation of Friedman's TreeBoost has been used, called XGBoost. The fundamental idea of TreeBoost is to create an ensemble via boosting of decision trees of a fixed size. A TreeBoost model is built iteratively, adding one tree at the time beginning with constant prediction at round zero. The ensemble can be viewed in Eq. 4. Selecting which tree to add is done via gradient decent of an objective function, which consists of a loss part (squared loss) and a regularization part. Details of XGBoost can be found in [17].

$$H(\mathbf{x})_i^{(t)} = \sum_{k=1}^{t} h_k(\mathbf{x}_i) = H(\mathbf{x})_i^{(t-1)} + h_t(\mathbf{x}_i) \tag{4}$$

To asses feature importance of boosted trees, it is essential to understand how individual trees in the ensemble measure feature importance. Variable importance of an individual tree, $h()$, the estimated importance squared of a feature, j, in a CART tree is expressed in Eq. 5. Where $J-1$ is the number of non-terminal nodes. $1()$ is a binary indicator function which indicates if a non-terminal node splits on feature j and $\hat{i}_t^2$ is the empirical improvement in squared error of the split. Friedman generalized this idea to measure feature importance over the whole ensemble, of M tree, by arguing that feature importance for the ensemble could be estimated as the mean of the features importance of the individual trees, as shown in Eq. 6.

$$\hat{I}_j^2(h) = \sum_{t=1}^{J-1} \hat{i}_t^2 1(v_t = j) \tag{5}$$

$$\hat{I}_j^2 = \frac{1}{M} \sum_{m=1}^{M} \hat{I}_j^2(h_m) \tag{6}$$

4 Results

In this section, data pre-processing and feature set are described along with performance of the current model, the machine learning models and their respective variable importance indications. The section is finalized by comparing the feature importance found by the two models to each other.

4.1 Data Processing and Feature Selection

Beside reducing number of features from 847 to 712, additional reduction was performed by using Correlation-based Feature Selection (CFS) [4], which reduced the number of features to 23. The intersection of the feature set selected by CFS and the one selected by domain experts showed that two features were shared among both sets. The union of the two sets generated a feature set with 28 different features (23 from CFS, 7 from experts, but 2 features are common in both sets). The union set was employed by the learning algorithms. The features correlations are displayed in Fig. 3, where $\{x_1, x_2, \ldots, x_7\}$ are the features selected by domain experts and the rest are features selected by the CFS.

To evaluate the performance of the current model, all 231 samples (28 features) are used. While for the machine learning models a 50-50 split is used to create a training and testing set, respectively.

4.2 Performance of the Current Model

Figure 4 is produced by applying the current model to all available samples and subtracting the actual time taken before the brightness changes in the final paper. The figure shows the error of the current model in terms of frequency. A mixture of two normal distributions was used to produce the density function which approximates the histogram. A standard EM algorithm was used to estimate the distribution [18]. The fitted density function is described in Eq. 7.

$$f_{mix}(\mathbf{x}) = \frac{\lambda_1}{\sigma_1, \sqrt{2\pi}} e^{-\frac{(\mathbf{x}-\mu_1)^2}{2\sigma_1^2}} + \frac{\lambda_2}{\sigma_2, \sqrt{2\pi}} e^{-\frac{(\mathbf{x}-\mu_2)^2}{2\sigma_2^2}} \tag{7}$$

It is worth noting that the mode of the distribution is shifted to the right. The empirical bias of the distribution described in Eq. 7 can be calculated by

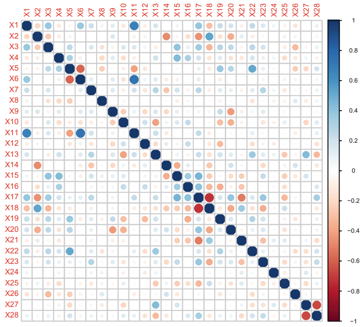


Fig. 3. Correlation between features selected by CFS and domain experts.

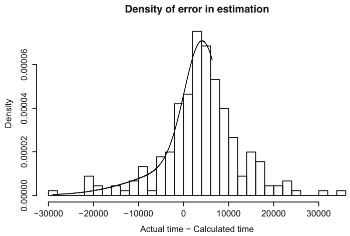


Fig. 4. Histogram showing error of current model, solid line is an estimated density function via a mixture of two normal distributions.

employing Eq. 8 to the estimated μ:s and λ:s. In the case of the current model, evaluated on the entire dataset, the value of μ_{mix} was 4130.

$$\mu_{mix} = \lambda_1\mu_1 + \lambda_2\mu_2 \tag{8}$$

The empirical variance of the current model was computed by Eq. 9. The standard deviation was $\sigma_{mix} = 8942\,\text{s}$.

$$\sigma^2_{mix} = \lambda_1\sigma^2_1 + \lambda_2\sigma^2_2 + \left[\lambda_1\mu^2_1 + \lambda_2\mu^2_2 - (\lambda_1\mu_1 + \lambda_1\mu_1)^2\right] \tag{9}$$

rMSE is used to compare the current model with the machine learning models, as they are trained with rMSE as evaluation criteria. rMSE is given in Eq. 10, where n is the number of samples, y is the target value and $\hat{y}$ is the target estimated by the model. The rMSE of the current model is 9568.

$$rMSE = \sqrt{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2} \tag{10}$$

4.3 Random Forest Results

To find optimal settings for an RF model leave-one-out cross validation was used in the parameter grid search. It was found that that parameter settings $p = 25$ and $K = 1000$ yielded the best performance. Figure 5 depicts one profile from the tuning of the parameter p, using fixed $K = 1000$. This figure suggests that $p \approx 25$ gives good generalization.

The RF model's performance was then measured against testing dataset. Figure 6 shows the error distribution as a histogram together with the estimated mixed model of normal distributions.

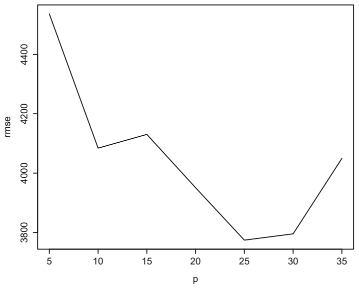


Fig. 5. Figure showing the RMSE for different values of p when training and evaluating the RF model using leave-one-out cross-validation, and $K = 1000$

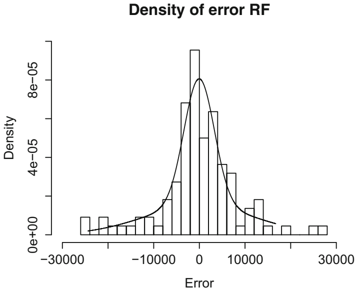


Fig. 6. Histogram showing the empirical error for the RF model on previously unseen data. A mixture of normal distributions was used to estimate the error frequency.

Using the methodology described earlier, the mean of the estimated mixed distribution was $\mu_{mix} = -45$ s, the empirical standard deviation was $\sigma_{mix} = 8533$ s and the $rMSE$ was 8147. It is evident that the RF model performs much better than the current model. It is interesting to investigate if RF utilizes other variables compared to the current model. However, due to random sampling of the subsets, different variable importance in terms of increased MSE, could differ between runs. To minimize such effect, 100 different RF models were created and evaluated in order to find the average variable importance. The results are shown in Fig. 9.

4.4 TreeBoost Results

The parameters of TreeBoost model was tuned in a similar manner to RF. TreeBoost requires two parameters to be tuned, the learning rate η, and the maximum depth of the trees D_{max}. The results from the grid search is visualized in Fig. 7. The performance of TreeBoost was noisy regarding its parameter settings, although a general trend is observed such that a low value of η yields a lower RMSE. The best setting was found at Depth = 7 and $\eta = 0.2$. When evaluating this model on the test set, an rMSE of 8176 is achieved. Similarly, a mixture of normal distributions was fitted to the errors depicted in the histogram of Fig. 8. The fitted distribution has $\mu_{mix} = 320$ s and $\sigma^2_{mix} = 8170$ s. The TreeBoost model, thus, has a slightly larger bias but lower variance compared to the RF model.

Results from the tuned TreeBoost model is shown in Fig. 10. Again, 100 models were trained and evaluated with regards to variable importance to negate effects of random sampling.

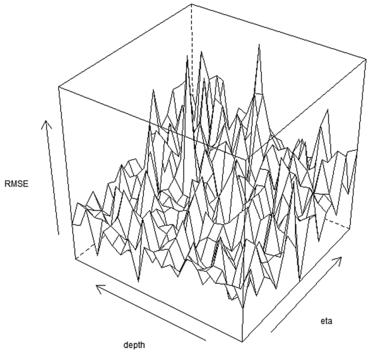


Fig. 7. Training error for different settings of TreeBoost, using leave-one-out cross validation.

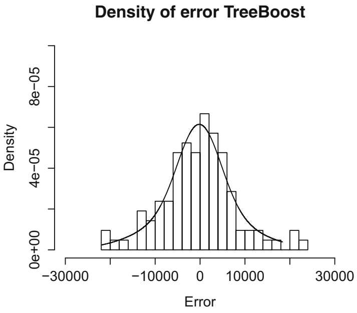


Fig. 8. A histogram showing the error distribution as well as a fitted mixture of normal distributions to describe the error.

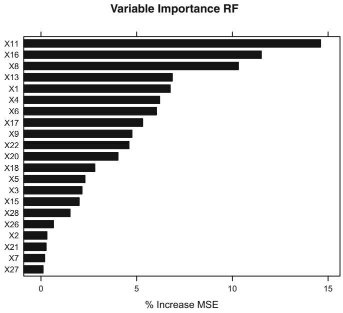


Fig. 9. Average variable importance for 100 different RF models. All using $p = 25$ and 1000 trees.

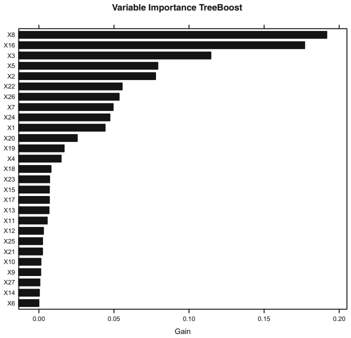


Fig. 10. Average variable importance for 100 different TreeBoost models. All using $\eta = 0.2$ and Depth = 7.

4.5 Comparing Variable Importance

From Figs. 9 and 10, it is clear that the different algorithms did not rank the feature identically. This is due to two facts. The algorithms do not measure feature importance in the same way and the algorithms uses two different methods to perform the ensemble. However, it is interesting to investigate if there is a correlation in the feature ranking. To measure this, Spearman's Rank Correlation was used. Given two rank vectors x and y, the rank correlation can be calculated by using Eq. 11. Where x describes the ranking of feature importance from RF and y describes the feature rankings from the TreeBoost model. Where n is the total number of features. The resulting Spearman's Rank Correlation are $\rho = 0.3218$, and $p = 0.0952$. These values suggest that the hypothesis that there is a weak correlation between the ranking of the two models is not rejected with an alpha-level of 0.1.

$$\rho = 1 - \frac{6 \sum (x-y)^2}{n^3 - n} \tag{11}$$

In Table 1 presents the top most important features as selected by the learning models together with those selected by the experts. Features selected by the models are those which gives 50 % increase/gain compared that given of the maximum important feature. Due to confidential reasons on behalf of the paper plant, numerical values are not provided but instead their physical interpretations are given.

Table 1. Top most important features found by both machine learning models, as well as the features selected by domain experts.

Feature	Type	Unit	Origin
X_1	Flow	m^3/h	Expert, RF
X_2	Level	%	Expert, TB
X_3	Level	%	Expert, TB
X_4	Flow	m^3/h	Expert
X_5	Flow	m^3/h	Expert, TB
X_6	Flow	m^3/h	Expert
X_7	Level	%	Expert
X_8	Concentration	%	RF, TB
X_{11}	Volume	m^3	RF
X_{13}	Level	%	RF
X_{16}	Voltage	kV	RF, TB

5 Discussion

It is interesting that the set of features found by CFS had an overlap with those selected by domain experts. This is promising since CFS could select any among the 712 features. Since the final pruned dataset (28 features) were created by taking the union between the current dataset and the one selected by CFS, it was expected that some of the variables would have high correlation between each other.

When comparing the feature importance rankings from the two machine learning models, indications of correlations are evident. It was found that X_{11} had high correlation with both X_1 and X_6. This is an interesting outcome as X_{11} were selected as the most important feature by the RF model, which in line with what the domain experts had deemed important for the system. On the other hand, TreeBoost did not select X_{11} as an important predictor and selected X_1 instead. Suggesting that both machine learning algorithms have an implicit agreement regarding the importance of these features. Since features are correlated, ranking them directly is hard. It was, therefore, encouraging to observe that the two different methods had an agreement in the ordering of the feature importance.

Despite the fact that only half of the samples were used for training, the machine learning models were able to outperform the current model for unseen data, with regards to bias, variance as well as rMSE. It has been seen that the machine learning models were able to drastically reduce the bias compared to the current model. This is expected since the current model does not try to actively predict the time of a changeover, but only the time it takes for bleaching to take effect. Unfortunately, it has not been possible to segregate the two time parts out of the target variable. However, given the assumption that changeovers are relatively constant in time, it is clear that the machine learning models were able to produce a significantly lower variance compared to the current model. This showed that, with the help of the extended feature set, it is possible to increase the performance when estimating the bleach time compared to the current model.

A feature that seems promising for future investigation is the feature X_8. What this feature describes is the new concentration of bleaching agent after the bleaching process is initiated. It is reasonable that such information could have predictive capabilities in how long the actual bleaching will take. X_8, as a reminder, was classified as the topmost important feature by the TreeBoost model and third most important by the Random Forest model. As this features is not used by the current model, it seems feasible to extend it by utilizing X_8 in order to increase its predictive capability. This is something that for practical reasons is good, as the two machine learning models are rather large and not practical to implement in the current control system used by the paper manufacturer. Results from this work indicate that improvements in bleach time production is possible.

6 Conclusion

In this work, an exploratory analysis of feature importance for predicting bleach time in a large paper manufacturing plant has been conducted. Different machine learning techniques, RF and TreeBoost, have been used to model bleach time. It has been shown that machine learning algorithms were able to reduce the prediction's bias, variance as well as rMSE. Further, by using Spearman's Rank Correlation, it is suggested that the feature importance ranking from the different machine learning algorithms were correlated. The big take-away from this article a new feature, X_8, were identified as important factor in estimating bleach time. This feature describes the concentration of bleach agent after the bleach process had been initiated. Future work will aim to incorporating X_8 into the pre-existing model such that it achieves improved predictive performance while still being in a format which is suitable for the current infrastructure used at the paper plant.

References

1. Key statistics, European pulp and paper industry (2014). http://www.cepi.org/system/files/public/documents/publications/statistics/2015/Key%20Statistics%202014%20FINAL.pdf. Accessed 11 Apr 2016
2. Electricity and heat statistics, eurostat. http://ec.europa.eu/eurostat/statistics-explained/index.php/Electricity_and_heat_statistics. Accessed 11 Apr 2016
3. Karl, H., Yella, S., Dougherty, M., Fleyeh, H.: Machine learning algorithms in heavy process manufacturing. Am. J. Intell. Syst. **6**, 1–13 (2016)
4. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning, pp. 359–366. Morgan Kaufmann (2000)
5. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
6. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. **29**, 1189–1232 (2001)
7. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. In: Proceedings of the 23rd International Conference on Machine Learning, ICML 2006, pp. 161–168. ACM, New York (2006)
8. Laha, D., Ren, Y., Suganthan, P.N.: Modeling of steelmaking process with effective machine learning techniques. Expert Syst. Appl. **42**(10), 4687–4696 (2015)
9. Halawani, S.M.: A study of decision tree ensembles and feature selection for steel plates faults detection. Int. J. Tech. Res. Appl. **2**(4), 127–131 (2014)
10. Deng, H., Runger, G.C.: Feature selection via regularized trees (2012). CoRR, vol. abs/1201.1587
11. Tumer, K., Ghosh, J.: Error correlation and error reduction in ensemble classifiers. Connect. Sci. **8**(34), 385–404 (1996)
12. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth and Brooks, Monterey (1984)
13. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
14. Breiman, L.: Bagging predictors. Mach. Learn. **24**, 123–140 (1996)
15. Aslam, J.A., Popa, R.A., Rivest, R.L.: On estimating the size and confidence of a statistical audit. In: Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology, EVT 2007, p. 8. USENIX Association, Berkeley (2007)
16. Schapire, R.E.: The strength of weak learnability. Mach. Learn. **5**, 197–227 (1990)
17. Introduction to boosted trees. http://xgboost.readthedocs.org/en/latest/model.html. Accessed 11 Mar 2016
18. Van Dyk, D.A., Meng, X.L., Rubin, D.B.: Maximum likelihood estimation via the ECM algorithm: computing the asymptotic variance. Technical report (1994)

Trust Model of Wireless Sensor Networks Based on Shannon Entropy

Jun Hu[1(✉)] and Chun Guan[2]

[1] School of Software, Nanchang University, Nanchang,
Jiangxi, China
hujun@ncu.edu.cn
[2] School of Information Engineering, Nanchang University,
Nanchang, Jiangxi, China
guanchun@yeah.net

Abstract. Trust between nodes is the foundation of communication security in wireless sensor network. A new entire trust model of wireless sensor network is constructed in terms of comprehensive trust value of nodes. The direct trust value is deduced on basis of node trusted elements system, according to the Shannon entropy theory. Experiments show that trust model of wireless sensor network based on Shannon entropy can improve the stability, speed and security of networks efficiently.

Keywords: Shannon entropy · Wireless sensor networks · Trust model

1 Introduction

Sensor nodes of Wireless Sensor Networks (WSN) are always deployed in hostile region or severe environment which is difficult to maintenance safely via manual method, and gain, collect, process and transfer information of local area or research objects [1–5]. There are many uncontrollable and variable factors in the environment of nodes, such as temperature, moisture, wind force, and pressure etc. Due to existence of those factors, nodes of sensors are ease to be attacked and malfunctioned, and it can make whole networks to be anomaly, even to be broken down. For example, when sensor nodes are deployed into hostile military area and are caught by enemy, the data from our networks would be leaked and modified maliciously, even camouflage node would be connected into our networks. In another scenario, if sensor nodes are deployed into severe outside natural area, internal modules of nodes are easily malfunction and cause network to break down, owning to influence of severe environment. Besides the influence of environment, sensor nodes are possible to behave selfishly, cause error or date lose, and break the function or lifetime of networks, in consideration of its limitation of communication, memory, power, computation, and unbalanced distribution. All external factors and internal limitation to damage WSNs are named as attack of WSNs.

Generally, the source of WSNs' attack can be classified into external attack and internal attack [6]. External attack includes information interception, information listening, camouflage node, fake route distribution, fake information transfer, service rejection etc. Internal attack includes package discarding, information resend,

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 397–403, 2016.
DOI: 10.1007/978-3-319-48743-4_32

information steal, fake data distribution and data modification etc. About external attack, there are many effective approaches against it, including general access control, intrusion detection, data authorization, digital watermarking, key security system etc. However, internal attack is difficult to defend effectively in that its indiscoverable and undetectable features, which don't exist in external attack, invalidate most general key security techniques. Thus, information security and efficiency of networks are threatened seriously in terms of general key security systems are entirely malfunctioned under internal attack. In this scenario, as the complement of general key security systems, trust model and trust management technique of WSNs, which depend on nodes' trust relationship and trust degree to judge anomaly behavior, are critical research fields of WSNs' security. Therefore, establishment of nodes' trust relation is essential condition to evaluate networks' functionality, since operation and maintenance of WSNs do generally rely on the trust between sensor batch nodes.

2 Related Work

Nowadays, domestic and international scholars, who research WSNs trust [7–12], generally focus on basic principle, behavior of nodes, trust degree of nodes, confidential degree of nodes, structure of networks, and coordination of nodes etc. Fuzzy theory, Subject Logical theory, Bayesian theory and Uncertainty deduction etc. are leveraged to construct trust model. Moreover, trust issues of WSNs are integrated into other theories, such as Cloud theory, Rasch theory, Social Networks theory, Analytic Hierarchy Process, and Grey theory etc., and correlate trust models are constructed on the basis of those theories [13–20].

However, there are still various defects in those theories and models, including:

1. Complicate computation and evaluation of trust value. Massive and delicate parameters for computation need more memory and CPU of nodes, which burden nodes' loads and energy consumption heavily.
2. Historical Storage of computing parameters. There are enormous parameters in the communication between nodes, history information of which need to store in the nodes' memory for a long time.
3. Simplification and deletion of parts of parameters. There are some unnecessary parameters that need to be simplified or deleted in terms of loose couple for actual application or weak effect for trust results.
4. Inaccuracy of trust results. In consideration of effect of only one or few elements in trust values, it will separate the relation among elements of trust system. Moreover, manual configuration or prior configuration for element weights will cause inaccuracy of trust values' evaluation.

3 Trust Model Based on Shannon Entropy

In landmark paper published in 1948, entitled "A Mathematical Theory of Communication", C. E. Shannon addressed conception of entropy, which analyzes and evaluates quantity of information in message.

Generally, signals from information source are uncertain, so that uncertainty can be evaluated by probability of occurrence. Probability of occurrence is more, uncertainty is less; on the other side, probability of is less, uncertainty is more.

Function of uncertainty F is monotonic decreasing function of Probability P, and the uncertainty function of two independent signals occurred in same time equals to the sum of uncertainty function of each signal, $F(p_1,p_2) = F(p_1) + F(p_2)$, named Additivity. The function F, which can satisfy Additive property and Monotonic decreasing property, shall be logarithmic function, $F(p) = \log\frac{1}{p} = -\log p$.

During analysis of quantity in information, it is not just consideration of uncertainty of one signal's occurrence, but average uncertainty of all signal's occurrence from source. If there are n values in signals from source: $U_1 \cdots U_i \cdots U_n$, the probability of occurrence is $P_1 \cdots P_i \cdots P_n$, and occurrence of each signal is independent, the average uncertain degree of source is statistic mean E of each signal's uncertainty $F(p_i)$, named Entropy, is defined as $H(U) = E(F(P_i)) = -\sum_{i=1}^{n} P_i \log p_i$, where base of logarithm is 2, and unit is bit, U represent collection of all signals may occurrence.

In consideration of massive and complicate information from sensor nodes in WSNs, this paper combines Shannon entropy, fuzzy trust model and trust model of Social Networks to deduce trust value of nodes precisely.

3.1 Direct Trust Elements System of Nodes

Direct trust of nodes is influenced by various trust elements, so that direct trust elements system of nodes is the foundation and prerequisite of computation of direct trust. The choice of trust elements shall conform to the features of WSNs and Wireless Sensor. Therefore, trust elements are classified into trust elements of nodes communication, trust elements of node structure, and other elements.

(1) Trust elements of nodes communication, include speed of data process, indicator of data process, transmit power of node signal.

Speed of processing data is data which are processed for same task in unit time by nodes, is defined:

Speed of processing data = Amount of processing data/processing time.

Indicator of processing data is rate of transfer data to receive data for same task during a time cycle by nodes, is that:

Indicator of processing data = Amount of transferring data/Amount of receiving data.

Transmit power of node signal is transmit power of sensor node during data transmission.

(2) Trust elements of node structure, include variation of memory capacity and consumption of power.

Variation of memory capacity is variation of memory capacity from beginning to end of data transfer by node for same task, is that variation of memory capacity = memory capacity at end of data transfer – memory capacity at beginning of data transfer.

Consumption of power means the power consumption of node for same task in unit time, is that:

Consumption rate of power = power consumption of node/Processing time.

(3) Other elements, include success rate of mission accomplished, distance between nodes, amount of neighbor nodes, distortion rate of data, and count of node communication.

Success rate of mission accomplished is that:

Success rate of mission accomplished = amount of mission accomplished/amount of mission.

Distortion rate of data is discrepancy rate of receive data of present node to receive data of prior node. It is that:

Distortion rate of data = amount of discarding data in a node/amount of receiving data in prior node.

Count of node communication is that user can deliver command of append node to all nodes in networks, when a new node appears and is added by user. Otherwise, if the node appears suddenly, its trust will be evaluated by count of node communication.

3.2 Weight Computation of Trust Elements Based on Shannon Entropy

Evaluating Node collects related data of trust elements from evaluated node. Suppose there are m evaluated nodes, n trust elements, value matrix of trust elements is that:

$$\mathrm{X} = \begin{bmatrix} \mathrm{x}_{11} & \mathrm{x}_{12} & \cdots & \mathrm{x}_{1\mathrm{n}} \\ \mathrm{x}_{21} & \mathrm{x}_{22} & \cdots & \mathrm{x}_{2\mathrm{n}} \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{x}_{\mathrm{m}1} & \mathrm{x}_{\mathrm{m}2} & \cdots & \mathrm{x}_{\mathrm{mn}} \end{bmatrix},$$

where $x = (x_{ij})_{mn}$. In the case of a trust element, large fluctuation of evaluated node x_{ij} means that this node is anomaly, need to be marked as anomaly node and deleted from networks in order to less effect for evaluation matrix.

Steps of weight computation of trust elements based on entropy, is described as following:

(1) Normalize of matrix:

$$x_{ij}^{'} = \frac{x_{ij} - \min\{x_j\}}{\max\{x_j\} - \min\{x_j\}}$$

(2) Computing weight of trust element j for node i, is $p(x_{ij})$.

$$p(x_{ij}) = \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}}$$

(3) Computing entropy of trust element j:

$$e_j = -k \sum_{i=1}^{m} p(x_{ij}) \ln p(x_{ij})$$

(4) Computing discrepancy factor of trust element j, $d_j = 1 - e_j$. If discrepancy factor is less, the influence to trust element j is less. If discrepancy factor is bigger, the influence to trust is more, and element is more important and will set more weight during the computation of direct trust.

(5) Computing weight of element j:

$$w_j = \frac{d_j}{\sum_{j=1}^{n} d_j}$$

(6) Evaluating trust value of elements:

$$s_{ij} = w_j \times x_{ij}^{'}$$

4 Validation from Simulation Experiment

This paper compares suggested novel trust model and trust propagation algorithm with general trust model and algorithm via MATLAB simulation software, and analyzes con and pro of these models and algorithms. Configurations of experiment environment are as following:

(1) Deploy 200 node from Uniform distribution in area 100*100.

(2) Element attributes of nodes includes speed of data process, indicator of data process, transmit power of node signal, variation of memory, consumption of power, success rate of mission accomplished, distortion rate of data, and count of node communication, distance with other node, and count of neighbor nodes.

(3) All nodes are kept in initial state at beginning of experiment. After running of simulation, element attributes of nodes start to be variable.

(4) All nodes are kept in trust state at beginning of experiment. After running of simulation, untrusted nodes will be generated.

(5) Via 100 iteration of simulation, variation in practical scenario of WSNs will be simulated.

Critical subject of trust model is to simplify and speed computation of trust value, in order to reduce consumption of resource, such as power, memory capacity and bandwidth etc. Moreover, precision of trust value can be increased, and stability, speed, security of networks can be fortified.

From Fig. 1, trust values of nodes in suggested trust model increase rapidly at beginning of experiment 1. After that, the trust values of nodes in suggested trust model tend to be stable gradually, which means the networks are more stable. On the contrary, trust value of nodes in general trust model increase slowly. After that, more fluctuation and low level of trust value in general trust model happened during the procedure of simulation. Therefore, suggested trust model of WSNs based on entropy can improve the stability, speed and security of networks efficiently.

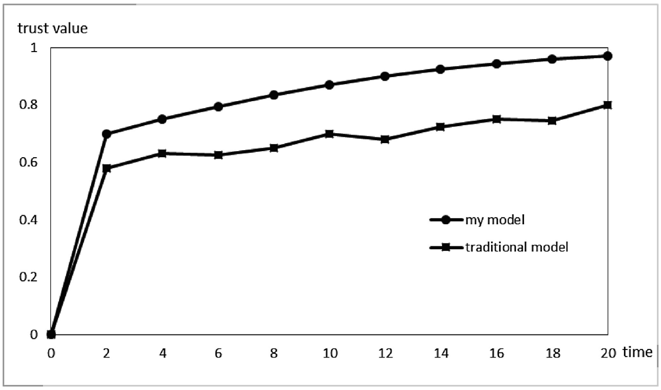


Fig. 1. Variation of trust value in trusted nodes

5 Conclusion

Information security and efficiency of WSNs are threatened seriously in terms of general key security systems are entirely malfunctioned under internal attack. In this scenario, as the complement of general key security systems, trust model and trust management technique of WSNs, which depend on nodes' trust relationship and trust degree to judge anomaly behavior, are critical research fields of WSNs' security. In this paper, a new entire trust model of wireless sensor network is constructed in terms of comprehensive trust value of nodes, which synthesize direct trust value and recommended trust value. The direct trust value is deduced on basis of node trusted elements system, according to the entropy of information theory. Simulation experiments show that the trust mode based on entropy can improve the stability, speed and security of networks efficiently. The following work will study how to compute recommend trust value according to the theory of social relational networks.

Acknowledgment. This work is supported by the Special Visiting Scholar Foundation for the Young Teacher Development Plan in Ordinary Universities of Jiangxi Province.

References

1. Han, G., Jiang, J., Shu, L., et al.: Management and applications of trust in wireless sensor networks: a survey. J. Comput. Syst. Sci. **80**(3), 602–617 (2014)
2. Aftab, M.U., Ashraf, O., Irfan, M., et al.: A review study of wireless sensor networks and its security. Commun. Netw. **7**(4), 172–179 (2015)
3. Ishmanov, F., Kim, S.W., Nam, S.Y.: A secure trust establishment scheme for wireless sensor networks. Sensors **14**(1), 1877–1897 (2014)
4. Alzaid, H., Alfaraj, M., Ries, S., Jøsang, A., Albabtain, M., Abuhaimed, A.: Reputation-based trust systems for wireless sensor networks: a comprehensive review. In: Fernández-Gago, C., Martinelli, F., Pearson, S., Agudo, I. (eds.) IFIPTM 2013. IAICT, vol. 401, pp. 66–82. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38323-6_5
5. Misra, S., Vaish, A.: Reputation-based role assignment for role-based access control in wireless sensor networks. Comput. Commun. **34**(3), 281–294 (2011)
6. He, D., Chen, C., Chan, S., et al.: Retrust: attack-resistant and lightweight trust management for medical sensor networks. IEEE Trans. Inf. Technol. Biomed. Publ. IEEE Eng. Med. Biol. Soc. **16**(4), 623–632 (2012)
7. Zhou, P., Jiang, S., Irissappane, A.A., et al.: Towards energy-efficient trust system through watchdog optimization for WSNs. IEEE Trans. Inf. Forensics Secur. (tifs) **10**(3), 613–625 (2015)
8. Feng, R., Che, S., Wang, X., et al.: Trust management scheme based on D-s evidence theory for wireless sensor networks. Int. J. Distrib. Sens. Netw. **2013**(1), 130–142 (2013)
9. Renjian, F., Xiaofeng, X., Xiang, Z., et al.: A trust evaluation algorithm for wireless sensor networks based on node behaviors and D-s evidence theory. Sensors **11**(2), 1345–1360 (2011)
10. Mármol, F.G., Pérez, G.M.: Providing trust in wireless sensor networks using a bio-inspired technique. Telecommun. Syst. **46**(2), 163–180 (2011)
11. Mármol, F.G., Pérez, G.M.: Trust and reputation models comparison. Internet Res. **21**(2), 138–153 (2011)
12. Shen, S., Huang, L., Fan, E., et al.: Trust dynamics in WSNs: an evolutionary game-theoretic approach. J. Sens. **2016**, 1–10 (2016)
13. Liu, A., Liu, X., Long, J.: A trust-based adaptive probability marking and storage traceback scheme for WSNs. Sensors (basel, Switzerland) **16**(4), 451 (2016)
14. Liu, L., Li, C., Jia, H.: Social Milieu oriented routing: a new dimension to enhance network security in WSNs. Sensors **16**(2), 247 (2016)
15. Vamsi, P.R., Kant, K.: Trust and location-aware routing protocol for wireless sensor networks. IETE J. Res., 1–11 (2016)
16. Dogan, G., Avincan, K.: MultiProTru: a Kalman filtering based trust architecture for two-hop wireless sensor networks. Peer-to-peer Netw. Appl., 1–14 (2016)
17. Ahmed, A., Bakar, K.A., Channa, M.I., et al.: A secure routing protocol with trust and energy awareness for wireless sensor network. Mob. Netw. Appl. **21**, 1–14 (2016)
18. Das, A.K.: A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks. Peer-to-peer Netw. Appl. **9**(1), 223–244 (2016)
19. Kaur, J., Gill, S.S., Dhaliwal, B.S.: Secure trust based key management routing framework for wireless sensor networks. J. Eng. **2016**(3), 1–9 (2016)
20. Dogan, G, Avincan, K, Brown, T.: DynamicMultiProTru: an adaptive trust model for wireless sensor networks. In: 2016 4th International Symposium on Digital Forensic and Security (ISDFS), pp. 49–52 (2016)

Assessing the Quality and Reliability of Visual Estimates in Determining Plant Cover on Railway Embankments

Siril Yella(✉) and Roger G. Nyberg

Department of Computer Engineering and Informatics,
Dalarna University, 78170 Borlänge, Sweden
{sye, rny}@du.se

Abstract. This study has investigated the quality and reliability of manual assessments on railway embankments within the domain of railway maintenance. Manually inspecting vegetation on railway embankments is slow and time consuming. Maintenance personnel also require extensive knowledge of the plant species, ecology and bio-diversity to be able to recommend appropriate maintenance action. The overall objective of the study is to investigate the reliable nature of manual inspection routines in favour an automatic approach. Visual estimates of plant cover reported by domain experts' have been studied on two separate railway sections in Sweden. The first study investigated visual estimates using aerial foliar cover (AFC) and sub-plot frequency (SF) methods to assess the plant cover on a railway section in Oxberg, Alvdalsbanan, Sweden. The second study investigated visual estimates using aerial canopy cover method on a railway section outside Vetlanda, Sweden. Visual estimates of the domain experts were recorded and analysis-of-variance (ANOVA) tests on the mean estimates were investigated to see whether if there were disagreements between the raters'. ICC(2, 1) was used to study the differences between the estimates. Results achieved in this work indicate statistically significant differences in the mean estimates of cover ($p < 0.05$) reported by the domain experts on both the occasions.

1 Introduction

Presence of vegetation on and alongside railway tracks is a serious problem. Vegetation on the tracks reduces the elasticity of the ballast and increase water retention eventually contributing to the deterioration of wooden sleepers. Vegetation alongside railway tracks (especially in curves and level crossings) severely challenges visibility, as a result of which, trains have to be slowed down. Proper control and maintenance is therefore necessary to ensure smooth operational routines. Inspections aimed at measuring vegetation on and alongside railway tracks and embankments in Sweden (and elsewhere in the world) are currently performed manually by a human operator. Such inspections are, to a large extent are carried either by visually inspecting the track on-site, or by manually looking at video clips collected by maintenance trains. A decision concerning the condition is given by the inspector and is normally only explicit in cases of poor condition, for which the inspector recommends further maintenance action. Manually

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 404–410, 2016.
DOI: 10.1007/978-3-319-48743-4_33

inspecting vegetation is slow and time consuming. Such inspection routines require that the maintenance personnel have extensive knowledge of the plant species to be able to recommend appropriate maintenance action. Further, preservation of the ecology and maintaining its diversity are yet other important issues [1–5]. The aim of this study is to investigate and assess the quality and reliability of such manual assessments; more specifically to compare the visual estimates (VE) of plant cover reported by the domain experts to be able to evaluate disagreements (if any).

Previous investigations on the problem reported significant difference between the raters' [6, 7]. Such work has mainly investigated the observers' ability to assess plant cover from images acquired on railway tracks to verify proof of concept. Results achieved from the above work indicate that seven out of the nine ANOVA tests conducted in this study have demonstrated significant difference in the mean estimates of cover ($p < 0.05$). The current article aims to extend the aforementioned work by carrying out cross investigations between domain experts on site as opposed to assessing vegetation from photographs. Work reported in this article is part of a major research project aimed at automating the process of detecting vegetation on railway embankments. A good description of the research project aimed at automatic vegetation detection on railway embankments is out of the scope of this article but could be found elsewhere [5, 8, 9]. Study of the prevalent differences between domain experts (if any) is therefore necessary to be able to advocate automatic procedures in favour of slow and time consuming manual routines.

The rest of the paper is organised as follows. Section 2 presents methodology; a brief introduction to the methods is also provided for the benefit of the readers unfamiliar with the methods. Section 3 presents results of the visual estimates reported by the different raters'. The paper finally presents concluding remarks.

2 Data Acquisition and Methodology

Studies aimed at investigating the VE of plant cover on railway embankments were carried out on two separate sites as follows.

2.1 On-Site Visual Estimates from Alvdalsbanan, Oxberg, Sweden

The first set of estimates was collected along the Alvdalsbanan railway track in Oxberg, Sweden. Two domain experts visually estimated the total plant cover of woody plants, herbs and grass separately (in %) using aerial foliar cover (AFC) and sub-plot frequency (SF) methods. In the context of assessing vegetation AFC is the area of ground covered by the vertical projection of the aerial portions of the plants. Small openings in the canopy and intra-specific overlap are excluded. In contrast, SF is a measure of the number of sub-plots that contain the target species. A good discussion concerning the methods could be found elsewhere [10]. At this stage it is worth mentioning that the raters' had long standing experience in estimating plant cover within the railway domain. Note that VE of mainly woody plants were reported along the Alvdalsbanan due to their dominant presence along the track. All the estimates were made by within a sample area of 1 × 1 m with the assistance of a boundary of a square meter (a.k.a. grid). The grid consisted of a sub-plot frame quadrat where each sub-plot measured 10*10 cm (see Fig. 1).

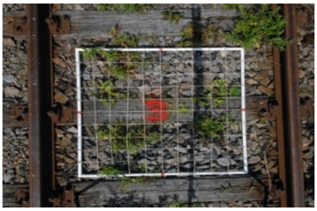


Fig. 1. Square meter grid with sub-plots

Different VE per plot were reported from a total of five plots in Oxberg as follows:

1. VE of the total cover using the AFC (no grid)
2. VE of the total cover using AFC and the square meter grid
3. VE of the woody plants using AFC and the square meter grid
4. VE of total cover using SF and the square meter grid
5. VE of woody plants cover using SF and the square meter grid

2.2 On-Site Visual Estimates from Vetlanda, Sweden

The second set of VE was collected from a section outside Vetlanda, Sweden. Three domain experts provided estimates on-site in 12 out of a total 179 sample areas. Twelve sample areas were selected by a systematic sampling method in which the starting position (of the first sample area) was chosen at random, and every eighth sampling area was assessed accordingly. Each sample area was represented by a rectangular area comprising of five ballast areas in between six sleepers on a railway track. VE in this particular case were reported using only the aerial canopy cover (ACC) method. ACC is the area of ground covered by the vertical projection of the outermost perimeter of the natural spread of foliage of plants, also known as the convex hull. Small openings within the canopy were included and in situations when it was practically impossible to identify individual plants, raters' agreed (in prior) to report such plant clusters as one plant.

At this stage it is worth mentioning that the railway section in this particular case was investigated twice in June and August 2013; to study the effect of a vegetation management routine that was carried out in between the sessions (see Table 2). The plant cover reported before and after the vegetation management routine (mainly herbicide treatment), expressed in percentage is presented in Table 2.

Note that no time limits were applied for gathering VE and the raters' reported their estimates independently on both sessions. All the observations were recorded and their mean estimates were computed for further analysis (see Table 1). Before proceeding any further it is worth mentioning that all the data was log10 transformed for all further parametric analysis. This is because a preliminary visual analysis of the raters' mean and median histogram plots has indicated an irregular, positively skewed distribution. The fact that log10-transformation makes a positively skewed data distribution less skewed justifies our choice [9].

Table 1. On-site visual estimates of plant cover (in %) at Oxberg, Alvdalsbanan, Sweden

Method	Rater	Plot 1	Plot 2	Plot 3	Plot 4	Plot 5	Mean diff. between the raters'
VE of total cover using AFC without a grid	Rater A	50	55	40	25	15	16
	Rater B	45	20	15	15	10	
	Relative difference between the raters	5	35	25	10	5	
VE of total cover using AFC and a grid	Rater A	62	37	16	15	13	5.6
	Rater B	52	28	20	18	15	
	Relative difference between the raters	10	9	4	3	2	
VE of woody plants cover using AFC and a grid	Rater A	30	21	15	14	9	2.2
	Rater B	25	22	13	13	7	
	Relative difference between the raters	5	1	2	1	2	
VE of total cover using SF and a grid aid	Rater A	86	59	46	61	47	17.8
	Rater B	94	81	68	73	72	
	Relative difference between the raters	8	22	22	12	25	
VE of woody plants cover using SF and a grid aid	Rater A	64	49	43	51	39	8.4
	Rater B	34	52	49	50	41	
	Relative difference between the raters	30	3	6	1	2	

Table 2. On-site visual estimates of plant cover at Vetlanda, Sweden

	(%) cover in June	(%) cover in August
Mean	12.89	2.6
Std. deviation	1.55	1.8
Max.	29	7
Min.	4	0

Analysis-of-variance (ANOVA) tests were tried and tested to investigate whether if there were differences between the estimates i.e. test the null hypothesis (H0) to check whether if the means of estimates are equal between the raters'. Density plots of the residuals obtained from the log10 transformed data were approximately normally distributed again justifying our choice of the ANOVA test. Intra-correlation coefficient (ICC) was used to be able to assess inter-rater reliability. Inter-rater reliability is the degree of agreement (a.k.a. ratings) between the raters' by comparing the variability of different ratings of the same subject with the total variation across all ratings and all subjects. The ICC coefficient can theoretically vary between 0 and 1.0, where an ICC value of 0 indicates no agreement whereas an ICC value of 1.0 indicates perfect agreement/reliability. A complete discussion of the classes is out of the scope of this article but could be found elsewhere [11]. In this particular article, ICC (2, 1) class was chosen (Eq. 1.).

$$ICC(2,1) = \frac{var(\beta)}{var(\alpha) + var(\beta) + var(\varepsilon)} \tag{1}$$

In addition to the ICC (2,1) method the Krippendorff's α was calculated using Eq. 2; where D_0 is the observed disagreement and D_e is the expected random disagreement.

$$\alpha_{Kripp} = 1 - \frac{D_0}{D_e} \tag{2}$$

3 Results and Discussion

Differences in the VE between the raters' were computed using ANOVA test and their reliability was assessed using the ICC (2,1) and the Krippendorff's α coefficient. Results achieved from the investigation at Oxberg, and Vetlanda have been tabulated in Tables 3 and 4 for the sake of simplicity.

Table 3. Inter-rater reliability between the visual estimates reported by the two raters' in Oxberg, Alvdalsbanan, Sweden

Method	Reliability ICC(2,1)	ICC(2,1) p-value	Krippendorrf's α
VE of total cover using AFC (no grid)	0.42	0.094	0.291
VE of total cover using AFC (grid aid)	0.94	0.0037	0.93
VE of woody plants cover using AFC (grid aid)	0.94	0.0012	0.935
VE of total cover using SF (grid aid)	0.46	0.016	0.283
VE of woody plants cover using SF (grid aid)	−0.58	0.83	−0.354

Table 4. Inter-rater reliability between the visual estimates reported by the three raters' in Vetlanda, Sweden

Method	Reliability ICC(2,1)	ICC(2,1) p-value	Krippendorrf's α
VE of total cover using ACC in June	0.53	$3.9 * 10^{-7}$	0.05
VE of total cover using ACC in August	0.51	$3.15 * 10^{-6}$	0.05

VE from Oxberg, Alvdalsbanan, Sweden indicate that better estimates were observed when the raters' had used the grid while estimating vegetation cover. Reliability coefficients while assessing the total cover and woody plants cover using AFC method assisted by a grid were relatively better. Reliability coefficients while assessing the woody plants cover using SF method assisted by a grid was in between moderate to poor and can be attributed to the high mean difference (17.8 %) in the VE for that trial. See Table 1. The reliability coefficient values of the VE while assessing the woody plants cover using SF method assisted by a grid were all negative. This indicates that the raters' estimates were worse than random. In particular remarkable differences have been observed when the raters' assessed the first plot. Post VE interviews revealed that individual interpretations as of how to assess bigger woody plants using SF methods led to the huge differences [9].

VE from Vetlanda, Sweden obtained in June and August 2013 were investigated using two one way ANOVA tests. Results achieved in both the cases showed a statistically significant difference between the three domain experts estimates ($p < 0.05$); indicating that the raters' disagreed on both the occasions. It was not in the interest of this investigation to identify which raters' differed from the others. Reliability of the raters' was assessed using ICC (2,1); see Table 4. ICC2 coefficient values in the current case could be considered as showing moderate reliability for a single rater i.e., how accurate a single rater would be if they made the estimates on their own [9].

4 Conclusions

Current day vegetation assessments within railway maintenance are (to a large extent) carried out manually; either through visual inspection onsite or by looking at video clips collected by maintenance trains. The overall objective of the study is to expose the unreliable nature of the (slow and time consuming) manual vegetation inspection regime in favour an automatic approach. The quality and reliability of such manual assessments have been investigated for the purpose by studying the visual estimates (VE) of plant cover reported by domain experts' on-site on two separate railway sections in Sweden. The first study investigated VE of domain experts using aerial foliage cover (AFC) and sub-plot frequency (SF) to assess the plant cover on a railway section in Oxberg, Alvdalsbanan, Sweden. A second study investigated VE of domain experts on a railway section outside Vetlanda, Sweden. VEfrom two separate occasions (in June and August 2013) were recorded using the aerial canopy cover method (ACC) on the same track with maintenance routine carried out in between the assessments.

VE of raters' were recorded and analysis-of-variance (ANOVA) tests on the mean estimates were investigated to see whether if there were disagreements between the raters'. ICC(2, 1) was used to study the differences between the estimates. Results achieved through ANOVA and ICC(2,1) tests clearly indicate that VE of plant cover are quite unreliable thereby suggesting the need for an automatic approach. Results achieved also highlight the importance of a well-defined protocol be presented to the personnel (in prior) to reduce systematic errors as a result of misinterpretation while assessing vegetation cover. There are other areas within the domain of railway maintenance that are (to this day) heavily reliant on manual inspections to ensure smooth operations. It would be interesting to extend the work further by investigating other immediately relevant areas and report if similar differences persist.

References

1. Hulin, B., Schussler, S.: Measuring vegetation along railway tracks. In: Proceedings of the IEEE Intelligent Transportation Systems Conference, pp. 561–565 (2005)
2. Banverket: Vegetation Maintenance Manual, Bvh 827.1, Original title in Swedish: Handbok om vegetation (2000)
3. Banverket: Vegetation Maintenance Requirements, Bvh 827.2, Original title in Swedish: behovsanalys infor vegetationsreglering (2001)
4. Banverket: Safety Inspections Manual, Bvf 807.2, Original title in Swedish: sakerhetsbesiktning av fasta anlaggningar (2005)
5. Yella, S., Nyberg, R.G., Payvar, B., Dougherty, M., Gupta, N.: Machine vision approach for automating vegetation detection on railway tracks. J. Intell. Syst. **22**(2), 179–196 (2013). ISSN: 2191–026X
6. Yella, S., Nyberg, Roger, G., Gupta, Narendra, K., Dougherty, M.: Reliability of manual assessments in determining the types of vegetation on railway tracks. In: Wang, J., Cellary, W., Wang, D., Wang, H., Chen, S.-C., Li, T., Zhang, Y. (eds.) WISE 2015. LNCS, vol. 9419, pp. 391–399. Springer, Heidelberg (2015). doi:10.1007/978-3-319-26187-4_37
7. Nyberg, R.G., Yella, S., Gupta, N., Dougherty, M.: Inter-rater reliability in determining types of vegetation on railway track beds, accepted for publication. In: The 3rd International Workshop on Data Quality and Trust in Big Data in Conjunction with the 16th International Conference on Web Information Systems Engineering (WISE), Miami, USA (2015)
8. Nyberg, R.G., Gupta, N.K., Yella, S., Dougherty, M.: Machine vision for condition monitoring vegetation on railway embankments. In: Proceedings of the 6th IET Conference on Railway Condition Monitoring (RCM 2014), Birmingham, UK (2014)
9. Nyberg, R.G.: Automated condition monitoring of vegetation on railway track beds and embankments, Ph.D. thesis, Edinburgh Napier University, UK (2015)
10. Coulloudon, B., Eshelman, K., Gianola, J., Nea, H.: Sampling vegetation attributes. Interagency Technical Reference BLM/RS/ST- 96/002+1730, Bureau of Land Management's National Applied Resource Sciences Center, Bureau of Land management. National Business Center. BC-650B. P.O. Box 25047. Denver, Colorado 80225–0047 (1999)
11. Shrout, P., Fleiss, J.: Intraclass correlations: uses in assessing rater reliability. Psychol. Bull. **86**, 420–428 (1979)

Community-Based Message Transmission with Energy Efficient in Opportunistic Networks

Sheng Zhang[1](✉), Xin Wang[1], Minghui Yao[1], and William Wei Song[2]

[1] School of Information Engineering, Nanchang Hangkong University, Nanchang, China
{zwxzs168,wx1991210}@126.com, hiyaominghui@163.com
[2] Business Intelligence and Informatics, Dalarna University, Borlänge, Sweden
wso@du.se

Abstract. An Opportunistic Networks is a wireless self-organized network, in which there is no need to build a fixed connectivity between source node and destination node, and the communication depends on the opportunity of node meeting. There are some classical message transmission algorithms, such as PRoPHET, MaxProp, and so on. In the Opportunity Networks with community characteristic, the different message transmission strategies can be sued in inter-community and intra-community. It improves the message successful delivery ratio significantly. The classical algorithms are CMTS and CMOT. We propose an energy efficient message forwarding algorithm (EEMF) for community-based Opportunistic Networks in this paper. When a message is transmitted, we consider not only the community characteristic, but also the residual energy of each node. The simulation results show that the EEMF algorithm can improve the message successful delivery ratio and reduce the network overhead obviously, in comparison with classical routing algorithms, such as PRoPHET, MaxProp, CMTS and CMOT. Meanwhile the EEMF algorithm can reduce the node's energy consumption and prolong the lifetime of network.

Keywords: Opportunistic networks · Message transmission · Community characteristic · Energy efficient

1 Introduction

An Opportunistic Networks (ON) is a network of wireless connected nodes. Nodes may be either mobile or fixed. The network topology may change due to node mobility or node activation and node deactivation. There is no a fixed connectivity between source node and destination node, and the communication depends on the opportunity of node meeting [1]. Due to the short-distance wireless mobile devices (such as smart phones, smart bracelets, Apple Watches, iPads, etc.) are widely used, the direct communication and data sharing for each other are becoming more and more convenient [2]. The typical applications of Opportunistic Networks are booming, such as the pocket

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 411–423, 2016.
DOI: 10.1007/978-3-319-48743-4_34

switched networks (PSN) [3], the mobile vehicular networks (VN) [4], and the wireless sensor networks (WSN) [5].

In Opportunistic Networks, the communication among nodes shows intermittent connectivity due to the node's moving. Therefore the Opportunistic Networks only depends on the encounter opportunity among nodes to forward messages. Consequently, the "Storage-Carry-Forward" strategy is usually used to deliver messages in Opportunistic Networks. In addition, the nodes generally tend to congregate together according to social relations in Opportunistic Networks, show community characteristic. The node is much active in itself community, while it hardly move to other communities. There are just a few nodes which can visit other communities according to their interests, they are likely to set up the ties between different communities.

Another feather of Opportunistic Networks is that most of nodes supply with batteries. The power of a battery is usually limited, and charging is not convenient in some case. So it is very important to save energy. In this case, how to forward efficiently messages from source node to destination node in the community-based opportunistic networks is a huge challenge.

The remainder of this paper is organized as follows. Section 2 describes the exiting message transmission mechanism and energy efficient solutions. Section 3 introduces the community division and message buffer for saving energy. Section 4 shows the energy efficient message forwarding algorithm (EEMF) for community-based Opportunistic Networks that we propose in this paper. In Sect. 5, we set the simulation scenario and analyze the experiment results. And Sect. 6 concludes overall paper and lists future work.

2 Related Work

Many researches have done the considerable works on message forwarding in Opportunistic Networks. Due to the intermittent connectivity of nodes, the message forwarding or not is mainly based on the encounter probability between the nodes. The typical message forwarding algorithms based on probability are PRoPHET [6], and MaxProp [7]. These algorithms rely on the meeting opportunity between nodes to achieve the goal node of forward, so as to improve the delivery successful ratio. But these algorithms do not consider the community characteristic of Opportunistic Networks. Some research scholars considered the community characteristic and proposed corresponding algorithms, such as CMTS [8], CMOT [9], OSNLMTS [10], etc. But these algorithms mainly considered to improve the message delivery successful ratio and reduce the network latency, did not take into account the energy consumption of networks.

At present, the research on the node energy consumption in Opportunistic Networks had made some progress. The research is mainly two directions.

Firstly, it is mainly considered in the hardware layer. Let the node sleep according to a certain way, or lower the power of node's scan and receive to get the goal of energy conservation based on the environment.

The literature [11] takes sleep mode. It saves energy according to sleep mode. And designed a framework makes nodes in sacrificing a small amount of communication opportunity can save energy, on the basis of the framework effectively balance the energy saving and node connectivity problems.

The literatures [12, 13] design an energy-saving MAC scheme. They propose as a kind of MAC layer routing protocol for WSN, named S-MAC. In order to reduce energy consumption when the S-MAC listen the idle channel, node periodically sleeps. It forms a virtual cluster with neighbor nodes at the same time. In the node dormancy, neighbor nodes automatically in sync.

The literatures [14, 15] design moving plan of node to reduce energy consumption. In the range of communication, MULEs collects data from the sensors, stores and transmit messages to the nodes which close the MULEs by wireless.

Secondly, energy consumption is considered in software layer. It uses suitable routing algorithm to reduce the amount of data packet transmission, achieves the goal of energy saving.

The literature [16] propose IC-Routing algorithm. In the environment of natural disaster, node's mobility and encounter show the characteristics of periodic. We choose a more worthy trust path by evaluating routing delay and the transmission probability, and control the message copies to reduce the network overload.

In this paper, we build a community-based network model and propose an energy efficient message forwarding algorithm (EEMF) for community-based Opportunistic Networks. We assume that there are n nodes in Opportunistic Networks. And we set a threshold value δ, if the encounter probability of node i and node j is bigger than δ, node i and j are belong to the same community. After the network runs for a period of time, the n nodes are assigned into different communities according to the encounter probability each other. The EEMF includes two parts: intra-community forwarding and inter-community transmission. The former adopts multi-copy forwarding strategy according to the node residual energy and the counter probability between nodes within a community, while the latter selects optimal path between the connected communities according to nodes' transmission probability. As this scheme considers both the local community characteristic and the connectivity among communities in global network, meanwhile adopts energy efficient strategy, it would be feasible to achieve the optimal performance.

3 Community Division and Message Buffer

3.1 Community Division

The node moves in Opportunistic Networks and shows community characteristic. The nodes have a high encounter probability in the same community, while there is a low encounter probability in different community. When the encounter probability is greater than the threshold δ, the two nodes will be divided into the same community. It can effectively reduce the weak connection problem by setting the threshold.

The process of community division is described as following.

Firstly, we initialize the local community of node i as $C_i = \{i\}$.

Secondly, when node i and j meet together, we update the encounter probability as $p_{(i,j)}$.

Thirdly, if $p_{(i,j)}$ is bigger than the threshold δ, then put the node j into the community of node i.

Newman and Girvan [17] proposed modularity to measure the performance of community division. The modularity function Q is used to measure the quality of community division. The Q would be bigger, when the nodes in the same community have a strong relation and the nodes in different communities have a weak relation. The bigger of Q value, the better of the community division quality. Because of different value of threshold δ would have different number of community, so there is different community division result with different threshold value δ. We can select different threshold δ, compare the modularity function Q, would get the best threshold δ.

By experiment verification, the best threshold δ is 0.25 in this paper.

The above-mentioned method can get the best community division result. We don't need to know the number of community in the network in advance. And the location and size of community are changed with the node moving. It's more flexible, suitable to the real situation.

3.2 Message Buffer

If the copies of message are too many in the network, it will waste the network resource. We introduce ACK mechanism to eliminate redundant message copies. When a message gets to the destination node, the destination node immediately broadcasts a respond message of ACK into the network. When a node gets the respond message, compares the message ID with messages in message buffer, and deletes the message with the same ID. When the message's time to live (TTL) is over the threshold, delete it.

4 Message Transmission Strategies of EEMF

4.1 Intra-community Forwarding Strategy

The PRoPHET is a classical probability-based transmission algorithm, defines the delivery predictability to measure delivery probability metric between nodes. If the delivery predictability of node j is larger than that of node i which carries with messages, the node j can gain a copy of the messages from node i. Since nodes move in a community frequently, the encounter probability between nodes is high, there are large number of message copies in network. In this case, a great deal of unnecessary messages are forwarded, they waste a lot of network resources. Therefore we propose an improved PRoPHET algorithm for intra-community message transmission in this paper. We select one-hop nodes for destination node as relay nodes and consider the residual energy of node. This way ensures high delivery ratio and reduce redundant message copies with energy efficiency.

Encounter Probability Between Nodes. Each node holds an encounter probability vector to store encounter probability between nodes. Whenever node i encounters node j, the encounter probability should be updated according to the formula (1), where

$p_{init} \in [0, 1]$ is an initialization constant. This formula ensures that nodes have high delivery predictability when they are often encountered.

$$p_{(i,j)} = p_{(i,j)_{old}} + (1 - p_{(i,j)_{old}}) \times p_{init} \tag{1}$$

If node i does not encounter node j during a time interval, they are less likely to become good forwarders of messages to each other. As a consequence, the delivery predictability must age. The aging equation is shown in formula (2), where $\gamma \in [0, 1]$ is the aging constant, and k is the number of time units. The time unit can be different, and should be defined based on the average interval of nodes encounter within the community.

$$p_{(i,j)} = p_{(i,j)_{old}} \times \gamma^{k} \tag{2}$$

The simulation results shown in Sect. 5 reveal that $p_{init} = 0.75$ and $\gamma = 0.98$ are the most appropriate values.

The residual energy factor σ of node can be gotten by the following formula (3), where σ_j is the residual energy factor of node j, E_{j}^{c} is the current residual energy of node j, and $E_{\mathrm{j}}^{\mathrm{i}}$ is the initial energy of node j.

$$\sigma_j = \frac{E_{\mathrm{j}}^{c}}{E_j^{i}} \tag{3}$$

We define the forwarding probability for considering the residual energy of encounter probability. So, the forwarding probability from node i to node j is shown in formula (4), where λ is weighting factor.

$$p_{(i,j)}^{'} = \lambda p_{(i,j)} + (1 - \lambda)\sigma_j \tag{4}$$

Message Forwarding Process in a Community. Intra-community message forwarding depends on the forwarding probability of node. When two nodes encounter, the EEMF compares the forwarding probability, and the messages always forward to the node whose forwarding probability is larger. If a node forwards a message to another node, it does not delete the message. If a node relays a message, it stores and manages the message in accordance with the “first-in first-out” principle, until the TTL (the time to live) value expires or the message is transferred to the destination node.

Meanwhile, If the messages are forwarded to the destination node, an ACK packet that carries the ID of the received message is sent to the network. When a node receives the ACK packet, it will eliminate the redundant message copies based on the ACK information.

From the perspective of energy saving, this message forwarding method selects the node with the highest forwarding probability as a relaying node to ensure the reliability of delivery. And it selects only one-hop node as the relaying node to reduce the number of redundant copies in the network.

4.2 Inter-community Transmission Strategy

The core of inter-community message transmission is to find the optimal path from the source node community to the destination node community.

Community Transmission Probability. We reference the concept of community transmission probability in literature [9]. Each node holds a community transmission probability table which stores the transmission probability from the node to each community. The community transmission probability is divided into two categories: the accessible community transmission probability and the inaccessible community transmission probability. For the local community, the value of community transmission probability is 1. For the accessible community and the inaccessible community, the values of community transmission probability are calculated as following.

The accessible community transmission probability of a node is the probability that the node visits the accessible community, is calculated by formula (5). Where p_{ic_j} is the community transmission probability of node i visiting community c_j, C_a is the accessible community set of the node i, N_{ic_j} is the number which node i visits community c_j.

$$p_{\mathrm{ic}_j} = \frac{N_{ic_j}}{\sum\limits_{c_k \in C_a} N_{ic_k}} \tag{5}$$

The inaccessible community transmission probability can be calculated by the accessible community transmission probability and the encounter probability. We assume that there are three communities: c_x, c_y, and c_z. c_x is the local community of node i, c_y is the local community of node j and the accessible community of node i, and c_z is the accessible community of node j and the inaccessible community of node i. The scenario is shown in Fig. 1.

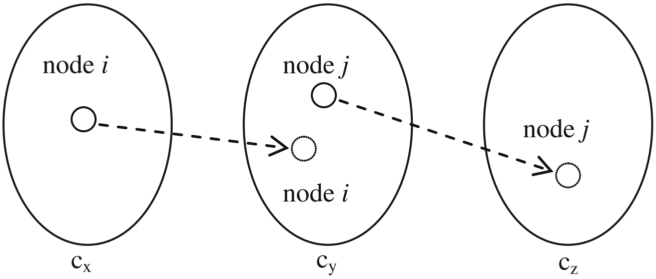


Fig. 1. The community path $(\mathrm{c}_x \rightarrow c_y \rightarrow c_z)$ is built by the node i and j

If the node i and j encounter each other, they exchange the community transmission probability table. A communication path from community c_x to community c_z (i.e. $\mathrm{c}_x \rightarrow c_y \rightarrow c_z$) is established when the node i encounter the node j. The node i has an opportunity to transfer messages from c_x to c_z through c_y.

Equation (6) shows the inaccessible community transmission probability of the node i to the inaccessible community c_z.

$$p_{ic_z} = p_{ic_y} \times p_{(i,j)} \times p_{jc_z} \tag{6}$$

Where p_{ic_y} indicates the accessible community transmission probability of the node i to the community c_y, and builds the community path $\mathrm{c}_x \rightarrow \mathrm{c}_y$. Likewise, $p_{\mathrm{j}c_z}$ indicates the accessible community transmission probability of the node j to the community c_z, and builds the community path $\mathrm{c}_y \rightarrow c_z$. $p_{(i,j)}$ indicates the encounter probability of node i and node j and provides the opportunity that the messages can transmit from the community c_x to the community c_z.

When we consider the energy efficient, we need replace the $p_{(i,j)}$ with the $p'_{(i,j)}$, then, the formula (6) change to formula (7).

$$p_{ic_z} = p_{ic_y} \times p'_{(i,j)} \times p_{jc_z} \tag{7}$$

Inter-community Message Forward Process. When messages are forwarded between communities, the community transmission probability of a node to the target communities is used to choose the best community communication path. Thus the node with the highest community transmission probability is often chosen as a relay node between communities, until the message is delivered to the target communities.

5 Simulations Scenario and Results Analysis

5.1 Simulation Scenario

In this paper, we use the ONE (Opportunistic Network Environment) to simulate, and compare with typical algorithms such as PRoPHET, CMTS, MaxProp and CMOT. The Fig. 2 shows the interface of the ONE simulation software.

Before the simulation begins, the pretreatment process of 10000 s completes the community division. The specific simulation parameters are set in Table 1.

5.2 Experimental Results and Analysis

Based on the above scenario, we compare the performance of five algorithms with different nodes average speed and messages TTL. The metrics include the successful delivery ratio, the average overhead and the residual energy. We still discuss the effects on energy consumption by selecting different λ value.

The effects of energy consumption by changing λ value To get the best energy saving, we need to seek the suitable value of λ. The simulation results are shown in Fig. 3, when λ is 0.7, 0.75, .80.8, 0.85, 0.9. When we set $\lambda = 0.85$, then network owns the best energy saving state. So we select $\lambda = 0.85$ in following experiments.

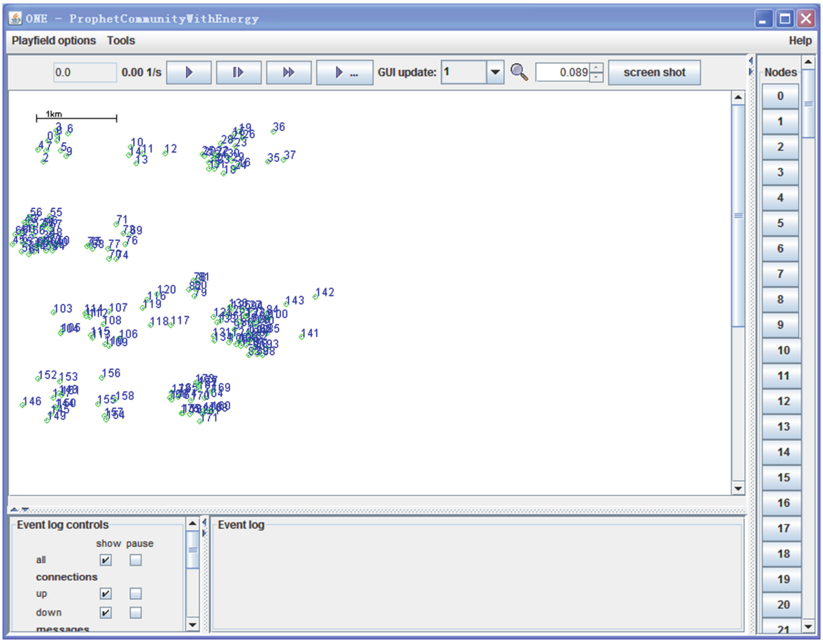


Fig. 2. The simulation interface of the ONE

Table 1. The parameters of simulation scenario

Category	Parameter (unit)	Values
Scenario features	Simulation time (s)	43200
	Threshold δ	0.25
	Simulation region (m^2)	8500 m × 8500 m
Community and node characteristics	Movement model	Community movement
	Initial energy (mA.H)	1000 k
	Energy in scan consumption (mA.H)	60
	Energy in transmit consumption (mA.H)	300
	Movement speed (m/s)	1～7
	Transmission rate (KB/s)	250
	The maximum transmission range (m)	30
	Cache size (MB)	10
	Wait time (s)	5～10
Data packet characteristics	Event generator	Message event generator
	Data packet size (MB)	0.5～1.5
	TTL (s)	1000/2000/4000/6000/8000/10000/12000
	The total number of data packets	1000

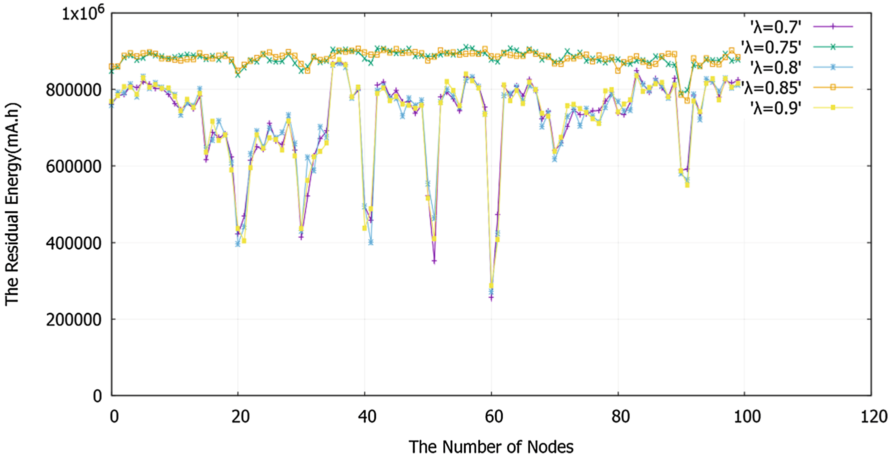


Fig. 3. The nodes energy consumption for different λ value

The effects in Different Average Speed of Nodes. We set the message TTL is 6000 s. As shown in Fig. 4, when the average speed increases, the successful delivery ratio improves for all algorithms. The EEMF and CMOT have the highest successful delivery ratio. When the average speed is less than 5 m/s, the successful delivery ratio increases linearly. Figure 5 shows the change of network average overhead, when the average speed increases. The EEMF and CMOT is lower network overhead than PRoPHET, CMTS, and MaxProp. Except for the MaxProp, other algorithms are not sensitive to the change of the average speed. When the average speed is greater than 5 m/s, the average overhead almost remain the same.

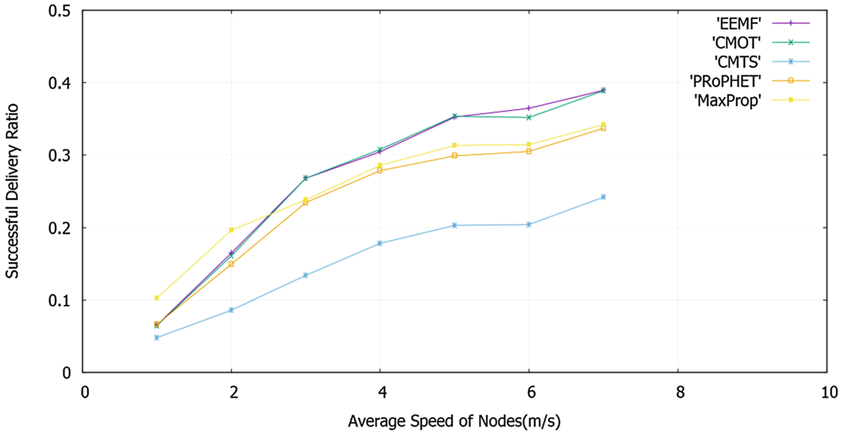


Fig. 4. Comparison of delivery ratio in different average speed of nodes

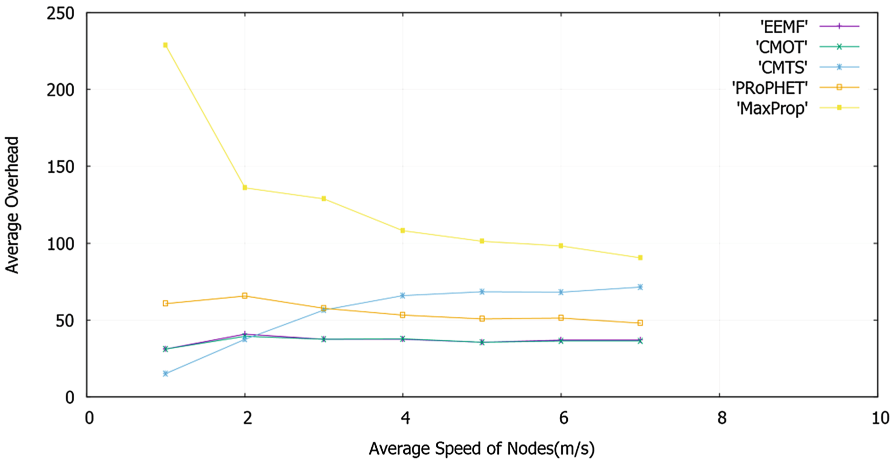


Fig. 5. Comparison of average overhead in different t average speed of nodes

The Effects in Different Messages TTL. We set the average speed of nodes is 5 m/s. Figure 6 shows that the successful delivery ratios of all algorithms are low when the message TTL is small. When the message TTL increases, the successful delivery ratio improves for all algorithms. When the message TTL is greater than 6000 s, all algorithm have slow increase of successful delivery ratio, except for CMTS. Meanwhile the CMTS is lower successful delivery ratio than that of other algorithms. Figure 7 shows that the EEMF and CMOT have lower overhead ratio than other algorithms. With the message TTL increasing, the average overhead of CMTS increase, while the average overhead of other algorithms decrease.

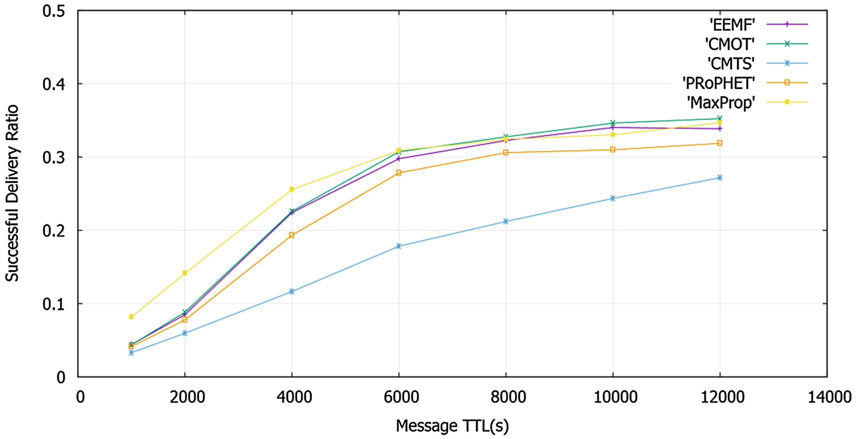


Fig. 6. Comparison of successful delivery ratio in different message TTL

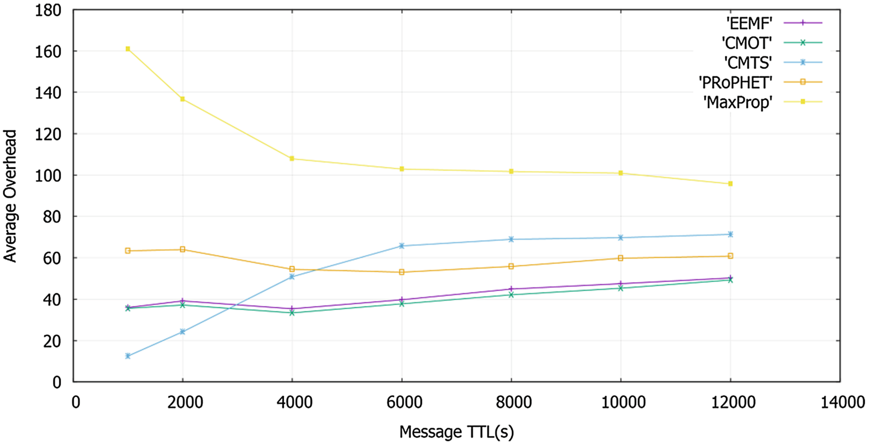


Fig. 7. Comparison of average overhead in different message TTL

The Comparison of Nodes' Residual Energy. We set the average speed of nodes is 5 m/s, and the message TTL is 6000 s. After the network runs in above different algorithms, the residual energy of all nodes are shown in Fig. 8. We can see that the EEMF and CMTS have the best energy saving effect.

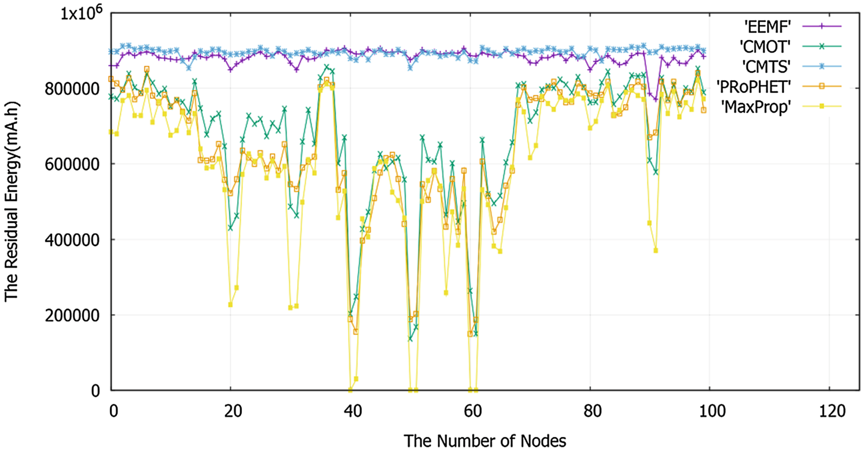


Fig. 8. The nodes residual energy after running different algorithm

6 Conclusion and Future Work

According to the community characteristic of nodes in Opportunistic Networks, We still consider the limited power supply by battery, propose an energy efficient message forwarding algorithm (EEMF) for community-based Opportunistic Networks.

The EEMF algorithm can reduce the node's energy consumption and prolong the life of network. Our major contributions are summarized as following. Firstly, we propose a new community division method, which is more suitable the community-based opportunistic networks. Secondly, on the basis of CMOT algorithm, we consider the energy saving strategy which can prolong the lifetime of network. The simulation results show that the EEMF and CMOT algorithm are better than the PRoPHET, MaxProp and CMTS algorithm in the successful delivery ratio and network overload. At the same time, the EEMF and CMTS algorithm are better than the CMOT, MaxProp and PRoPHET algorithm in energy saving effect. In short, the EEMF algorithm combines the advantages of other algorithms, improves the successful delivery ratio, reduces the network overload and energy consumption of nodes at the same time, so as to prolong the network lifetime.

Acknowledgment. This work is partially supported by the National Natural Science Foundation of China (61162002, 61661037), and the Jiangxi province National Natural Science Foundation of China (20151BAB207038).

References

1. Xiong, Y.P., Sun, L.M., Niu, J.W., Liu, Y.: Opportunistic networks. J. Softw. **20**(1), 124–137 (2009)
2. Wu, J., Xiao, M., Huang, L.: Homing spread: community home-based multi-copy routing in mobile social networks. In: 2013 Proceedings IEEE, INFOCOM, pp. 2319–2327 (2013)
3. Ma, C., Yang, J., Du, Z., Zhang, C.: Overview of routing algorithm in pocket switched networks. In: 9th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), pp. 42–46. IEEE Computer Society (2014)
4. Gaito, S., Maggiorini, D., Rossi, G.P., Sala, A.: Bus switched networks: an ad hoc mobile platform enabling urban-wide communications. Ad Hoc Netw. **10**(6), 931–945 (2012)
5. Hu, S.C., Wang, Y.C., Huang, C.Y., Tseng, Y.C.: Measuring air quality in city areas by vehicular wireless sensor networks. J. Syst. Softw. **84**(11), 2005–2012 (2011)
6. Lindgren, A., Doria, A., Schelén, O.: Probabilistic routing in intermittently connected networks. In: Dini, P., Lorenz, P., Souza, JNd (eds.) SAPIR 2004. LNCS, vol. 3126, pp. 239–254. Springer, Heidelberg (2004)
7. Burgess, J., Gallagher, B., Jensen, D., Levine, B.N.: MaxProp: routing for vehicle-based disruption-tolerant networks. In: 25th IEEE International Conference on Computer Communications, pp. 1–11. IEEE (2006)
8. Niu, J., Zhou, X., Liu, Y., Sun, L.: A message transmission scheme for community-based opportunistic network. J. Comput. Res. Dev. **46**(12), 2068–2075 (2009)
9. Zhang, S., Tan, P., Bao, X., Song, W.W., Liu, X.: Community-based message opportunistic transmission. In: Vogel, D., Guo, X., Linger, H., Barry, C., Lang, M., Schneider, C. (eds.) Transforming Healthcare Through Information Systems. LNISO, vol. 17, pp. 79–93. Springer, Heidelberg (2016)
10. Liu, Y., Gao, Y., Qiao, J., Tan, C.: Community-based message transmission scheme in opportunistic social networks. J. Comput. Appl. **33**(5), 1212–1216 (2013)

11. Feeney, L.M., Nilsson, M.: Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In: Proceedings - IEEE INFOCOM, vol. 3, pp. 1548–1557 (2001)
12. Singh, S., Raghavendra, C.S.: PAMAS – power aware multi-access protocol with signalling for ad hoc networks. ACM SIGCOMM Comput. Commun. Rev. **28**(3), 5–26 (1998)
13. Ye, W., Heidemann, J., Estrin, D.: An energy-efficient MAC protocol for wireless sensor networks. In: Global Telecommunications Conference, GLOBECOM 2005, vol. 3, pp. 1567–1576. IEEE (2008)
14. Ramiro, S., Stolwijk, C., Dougados, M., van Tubergen, A.: Data mules: modeling and analysis of a three-tier architecture for sparse sensor networks. Ad Hoc Netw. **1**(2–3), 215–233 (2003)
15. Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L.S., Rubenstein, D.: Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. In: International Conference on Architectural Support for Programming Languages and Operating Systems, vol. 37, pp. 96–107 (2002)
16. Uddin, M.Y.S., Ahmadi, H., Abdelzaher, T., Kravets, R.: A low-energy, multi-copy inter-contact routing protocol for disaster response networks. In: IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2009, pp. 637–645. IEEE (2009)
17. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E Stat. Nonlinear Soft Matter Phys. **69**(2 Pt 2), 026113 (2004)

A Multi-Semantic Classification Model of Reviews Based on Directed Weighted Graph

Shaozhong Zhang[1(✉)], William Wei Song[2], Minjie Ding[1], and Ping Hu[1]

[1] School of Electronic and Computer Science, Zhejiang Wanli University,
Ningbo 315100, Zhejiang, China
dlut_z88@163.com

[2] Information Systems and Business Intelligence,
Dalarna University, 79188 Borlänge, Sweden
wso@du.se

Abstract. Semantic and sentimental analysis plays an important role in natural language processing, especially in textual analysis, and has a wide range of applications in web information processing and management. This paper intends to present a sentimental analysis framework based on the directed weighted graph method, which is used for semantic classification of the textual comments, i.e. user reviews, collected from the e-commerce websites. The directed weighted graph defines a formal semantics lexical as a semantic body, denoted to be a node in the graph. The directed links in the graph, representing the relationships between the nodes, are used to connect nodes to each other with their weights. Then a directed weighted graph is constructed with semantic nodes and their interrelationships relations. The experimental results show that the method proposed in the paper can classify the semantics into different classification based on the computation of the path lengths with a threshold.

Keywords: Directed weighted graph · Reviews · Semantic classification

1 Introduction

Semantic analysis is a branch of natural language processing. It studies the meanings and characteristics of a given text through analyzing its vocabulary. In recent years, with the development of social network systems and increasing interactive activities performed by users, reviewing the web services and products has been considered and required over all aspects of various networks, including social, economic, political, and commercial networks. Particularly, as e-commerce products and websites have rapidly booming, users, including buyers, sellers, and go-betweens are increasingly required to provide their comments (i.e. reviews) on merchandises, products, services, and even business activities and behaviors. Undoubtedly, reviews have become an important factor in improving the quality of businesses, products, services, and even ways of doing businesses. However, although there are many researches on investigation of using user reviews to support businesses, to dig into the semantics (semantic and sentimental analysis) of the user reviews remains less touched. Therefore, it is significant to carry out the study of reviews semantic analysis of e-commerce and social networks.

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 424–435, 2016.
DOI: 10.1007/978-3-319-48743-4_35

Generally, given an entity, its semantics consists of five components, a semantics target, properties of the semantic target, meaning, semantic holders, and a time when the entity pertains the semantic. In previous studies, generally, researchers considered to use entity to represent semantic targets (objects) [1]. An entity can be a product, a service, a topic, a person, an organization, or an event. Usually, an entity has a hierarchical structure and is described as a set of properties. Each property of an entity at different levels has its own value [2]. Sentimental values usually represent some semantics of the entity, expressing the sentiment of an entity as a positive, negative, or neutral style. Of course, a higher intensity of the sentiments at different levels can also be defined to describe opinions at great details [3–5].

A general task of a sentiment analysis is, for a given review text, to mine and extract its entity semantics and characteristics. A key task of sentimental analysis is achieved through its name (or naming) entity [6–8]. With the entity naming recognition, we try to classify entities into different categories, to which similar entities belong [9, 10]. The characteristics and classifications of entities form attributes of the entities used to represent the sentiments of the entities and the research of semantic-sentimental classification analysis is the key issue to the obtainment of an overall sentiment of a given review text (or a review document). Conventional machine learning methods used for semantic categorization of entity sentiments include support vector machines (SVM), maximum entropy, and naive Bayesian classifier. [11–13]. These methods are mainly used for topic-based semantic classification. However, with the quick increase and great diversity of item reviews provided by various types of users, many problems occur when processing the sentiments of the user reviews, including (1) uncertainty – a review being less relevant to a given text; (2) short texts – being hard to determine the exact meaning without context; (3) randomness – review target may change; and (4) being not strictly comply with the integrity of information. Furthermore, a target entity included in a review text is not just one; there may be more than one target entity. The semantics of the given review text may differ from one entity to another. That is, in one same review, it may exhibit one sentiment to an entity while to another entity it may present another sentiment. Hence, it is the most important issue of how to distinguish different semantic classification of different entities in the same review. That is a so called multi-classification problem of semantic-sentimental analysis study.

2 Related Work

In recent years, extensive researches have been carried out in the field of semantic analysis and opinion mining to analyze the review textual contents from the social networks and the web information systems. The purpose of semantic analysis and opinion mining is to find out what a user's views, such as attitudes and emotions, are on a particular entity. The entities represent an individual, an event, or a theme. Currently, most of the methods, dealing with users' opinions and emotions, focus on the research of classification and similarity of text data [14]. The existing semantic classification methods are divided into supervised and unsupervised ones. In their report [9], the authors presented a machine learning algorithm based on distance monitoring, which divides the Twitter information into two types, positive and negative, and uses distant supervision to classify the sentiment of Twitter messages. In their paper [15], Li and Liu proposed a TF-IDF weighting method and apply it to the voting mechanism and importing term scores, which provide an acceptable and stable clustering result. This approach displays an investigation direction of positive and negative polarity classification. A rule-based sentiment polarity calculation method is applied for extracting sentiment features from Chinese reviews [16], based on a sentiment word dictionary to calculate the basic polarity of the sentiment features. The method also considers to judge the dynamic sentiment word's polarity and to adjust the polarity according to the context information. Pak and Paroubek proposed an automatic method to collect a corpus of positive and negative sentiments without human interference and a corpus of objective texts [17]. The size of the collected corpora can be arbitrarily large. The method performs a statistical linguistic analysis on the collected corpus to generate a sentiment classification. In their paper [18] the authors developed a sentiment analysis system for reviews with Chinese sentimental orientation. The system analyzes the problems of tendency of semantic content of the reviews based certain characteristics and following a particular categorization. They propose the concept of dependencies to identify reviews' sentimental orientation.

3 Sentiment Analysis Framework

3.1 Semantic and Objects

Definition (Semantics): A semantic is a quadruple, (g, s, h, t), where g is the semantic target, also called object, s is the semantic about the object, h is the semantic holder, and t is the time when the semantic was expressed. In practice, the object can often be decomposed and described in a structured manner with multiple levels, which greatly facilitate both mining of semantic s and later use of the mined semantic results [1].

Definition (Object Structure): An object o is a product, service, topic, issue, person, organization, or event. It is described with a pair, o: (T, W), where T is a hierarchy of parts, sub-parts, and so on, and W is a set of attributes of o. Each part or sub-part also has its own set of attributes [1].

The relationship between semantic and object is shown in Fig. 1.

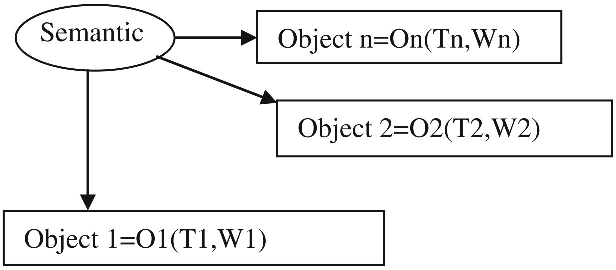


Fig. 1. Construction of semantic and objects

Look at Example 1. There is a review on a hotel on booking.com. “Loved the decor of the hotel and the location was perfect for shopping (1) and very close to the train station (2). Friendly and professional front desk staff (3). But it was very hot in our rooms because of the weather (4). I feel that this hotel needed air conditioning because my husband and I didn’t sleep any of the nights because of being so uncomfortable (5). Would visit this hotel again only in the cooler months (6). Also the breakfast wasn’t satisfactory (7). They kept running out of fruit and fruit salad - and for someone with allergies to other things, you rely on fruit (8).”

The semantic of the above can be decomposed into eight objects and corresponding attributes of the object and represented as pairs. This definition essentially describes a hierarchical decomposition of an object based on the part of relation. The root node is the name of the object, e.g., Review on Hotel. All the other nodes are parts and sub-parts, etc. A semantic can be expressed on any node and any attribute of the node.

Considering Example 2, in our example review above, the sentences (1), (2), (3) and (6) express positive opinions about the hotel. The sentences (4), (5), (7) and (8) express negative comments on the hotel. Clearly, one can also express semantics about parts or components of the hotel.

This object as a hierarchy of any number of levels needs a nested relation to represent it, which is often too complex for applications. The main reason is that since NLP is a very difficult task, recognizing parts and attributes of an object at different levels of details is extremely hard [1]. We use a directed graph to denote object and path for relation.

3.2 Directed Weighted Graph

In order to describe and extract semantic, emotional tendencies, we use a directed weighted graph to represent the structure of reviews text. We consider that the theory and method of graph structure is suitable for textual semantic analysis. Formally, a directed weighted graph is a set of triples. A triple is a collection of nodes, a set of links, and a collection of weights.

Definition (Directed Weighted Graph): A directed weighted graph is G defined as $G = <V, E, K>$, where V represents a set of nodes, E a set of edges, K a set of weights.

In the directed weighted graph, we consider to use the *markedness* and *orderliness* to describe the tags of semantic and emotion vocabulary appearing in a review text. For a given node v in the graph G, based on the order of the review text, we consider a (time) sequential relationship among other nodes connected to v by using the directed weighted graph. In the directed weighted graph G are there three types of nodes, i.e., text data nodes V^T, semantic and emotion tags nodes V^S, and sematic classification nodes V^C. So $V = \{V^T, V^S, V^C\}$. A text data node represents a text in the reviews. A semantic and emotion tag node corresponds to a particular emotion or specific semantic tags.

Definition (Links): In the space of nodes V of G, there are n nodes, $V_1, \ldots, V_n$. For any two nodes V_i and V_j, a directed link E_{ij} goes from V_i to V_j if $i < j$ (V_j appears after V_i), denoted to be $E_{ij} = V_i \rightarrow V_j$.

The value (weight) of a directed link between two nodes represents the strength of semantics and tendentious. The value can be calculated with the weights of a link. Directed links connect text nodes with tags nodes, tags nodes with others tags nodes, and tags nodes with semantic nodes. The strength of a connection is represented by a weight function of links.

Definition (Weighted Function): A weight function K_{ij} of links is used to compute the directed link weights of E_{ij} connecting any two nodes V_i and V_j, $i,j = 1, ..., n$.

With this method the semantics of a text can be represented by a path consisting of a set of weights and a series of tags and the strength of the semantics can be calculated by the weights in the path.

3.3 Semantic Classification Model

Basically, the idea of a semantic classification model is based on the directed weighted graph, i.e. $G = <\{V^T, V^S, V^C\}, E, K>$. The essential factors are described as follows. Firstly, given a review text data node, V^T, its properties are expressed as $V^T = <$ *ReviewerID, Data*>. Secondly, considering V^S as the semantic keywords, tags and other feature nodes extracted from a review text, we have $V^S \in <SignTagSet>$. Here *SignTagSet* is a known semantic lexicon library and contains two parts (signs), a positive part and a negative part. Thirdly, all the vocabularies are signs with certain scores. V^C denotes a set of semantic classification nodes. It represents different semantics and tendentiousness.

Definition (Semantic Classification): A semantic classification is defined as V^C, and the i^{th} element V_i^C in VC is defined to be:

$$V_i^C = (V_i^T, V_j^S, V_k^S, E_{ij}, E_{jk} | E_{ij}, E_{jk} \in E,)$$

Here V_i^C, a node set, is the *ith* classification; V_i^T is the node of the *ith* review text; V_j^S is the set of the *jth* semantic keywords in semantic lexicon; V_k^S is the kth tags node set of the meaning objects; E_{ij} connects the node V_i^T to the node set V_j^S, and E_{jk} connects the node set V_j^S to the node set V_k^S.

4 Weights and Semantic Classification Algorithm

4.1 Weights of K_{ij} Between Nodes in *G*

A weight on a link represents a frequency of some tags nodes in text, the tightness between tags nodes, and the similarity relationship between tags node and a particular semantic content node. The K_{ij} represents the weight of a directed link which is from a node V_i to another node V_j. We define a function $N(V_{ij})$ to be the number of the directed links from V_i to V_j. A junction-weight K_{ij} between two nodes and a sum of the junction-weights can be defined as:

$$K_{ij} = \sum_{V_j \in (N(V_i) \cap N(V_j))} N(V_{ij}) \text{ and } \sum\nolimits_{ij} K_{ij} \leftarrow \left| V(V^T, V^S, V^C) \right|$$

Algorithm (To compute the weights of K)

Input: the node set V_i^T of a review text and the set of semantic nodes extracted from the reviews V_j^S;

Output: All values of K that between the nodes;

Step 1: Initialization: $i = 1, j = 1, k = 1$;

Step 2: For each text node in V_i^T, calculate its frequency that its feature values appeared in the other tags node set, i.e. $N(V_{ij})$.

Step 3: Move to next tag node in V_j^S and $j = j + 1$;

Step 4: Return to Step 2, until searching all the tag nodes in V_j^S. For each tag node Calculate its number of frequencies of the junction from V_i^T;

Step 5: For a tag node in V_k^S calculate the frequency of characteristic value from V_k^S to another tag node marker, i.e. $N(V_{kj})$;

Step 6: Next tag node in V_k^S, $k = k+1$, return to Step 5 for all the nodes in V_k^S;

Step 7: For a text node of V_i^T, $i = i+1$, return to Step 2, for all the nodes in V_i^T;

Step 8: Compute the following formula $K_{ij} = \sum_{V_j \in (N(V_i) \cap N(V_j))} N(V_{ij})$ and $K_{kj} = \sum_{V_j \in (N(V_k) \cap N(V_j))} N(V_{kj})$

With the above algorithm we calculate the semantic closeness from the review text nodes to all the tag nodes, as well as all the other tag nodes. By setting different thresholds, we connect two nodes with a link with the values of path weights reaching a certain range, i.e. a directed link E_{ij} between V_i and V_j. Hence we construct a directed weighted graph with reviews text nodes and some semantic tag nodes.

4.2 Algorithm of Semantic Classification

The basic idea of semantic classification is to estimate the value of each path from a review text node to a tag node. Every weight on the path will calculate cumulatively. The objective is to identify a set of paths having their path length being not greater than a certain threshold. Each set of nodes on a path represents one kind of meaning or opinion of a review text, which can be as a classification of the reviews. In the actual calculation, using the path length to calculate the path weight is more convenient. The reciprocal of a path length is considered to be the weight of the path. The larger the value of weight is, the shorter the path length. Now our task is to find a list of different kinds of paths, with their sum of the path length being within a certain threshold. The nodes on one path give one similar semantic and can be classified in one category. By setting different thresholds, we obtain different paths lengths, thus generate a semantic classification of the reviews (i.e. different semantics or opinions).

We define the reciprocal of a weight as the length of a path from a node V_i to another node V_j, denoted $S_{ij} = S(V_i, V_j) = 1/K_{ij}$. Obviously, when a weight of a link is zero, i.e. $K_{ij} = 0$, the path length of the link $S(V_i, V_j) = \infty$. When the path length of a link (from one node to another) is shorter, the tightness between the nodes is higher. When searching the nodes that form all the paths, we can identify similar semantic content of some nodes linked together through a path. In this way, we can produce a series of paths whose nodes have different semantic similarities (i.e. classification of opinion semantics) by setting a range of thresholds for the path lengths. Of all these semantic classifications, the semantic class with the shortest path contains the review texts having closest semantics (the same or similar meaning). Semantic classification may vary when the range of path lengths is adjusted through the thresholds. This forms different types of semantic classification, termed Multi-Semantic Classification. The algorithm of Multi-Semantic Classification is given below.

Algorithm (To compute Multi-Semantic Classification)

Input: review text nodes V_i^T and tag nodes V_j^S; the weight value between nodes K_{ij} and K_{kj}.

Output: A collection of semantic classification:$V^C|Max(W_i|W_i = \sum E_{ij})$.

Step 1: Initialize the paths, $s \leftarrow V_i^T$, $R = \{s : V_i^T\}$, set the path length threshold.

Step 2: Select a tag node from V_j^S, and decide: If $S_{ij} = S(s, V_j) = 1/K_{ij} > \varepsilon$ (indicating that the tightness is smaller than the expectation), discard the node. Otherwise, if $S_{ij} <= \varepsilon$, add the node and continue.

Step 3: If the node V_j^S is already in the set R of path nodes the process has completed, go on to Step 2; if not, add V_j^S to R. Set the node V_j^S as the source node s, represented as $Q_{V_j^S}$, i.e. $R = R \cup \{V_j^S\}$, $Q_{V_j^S} \leftarrow s$.

Step 4: For each node V_k^S on the path from s to V_j^S, judge:

(1) If the distance of $s \rightarrow V_k^S$ is the shortest then add the path to s;

(2) If the V_k^S is an intermediate node of the path and the node is on the path from s to the node V_j^S, then delete V_k^S, and connect s to V_j^S;

(3) If the path from $V_k^S \rightarrow V_j^S$ is on the way of the path from s to V_j^S, i.e. $V_k^S \rightarrow V_j^S$ is a subset of $s \rightarrow V_j^S$, then add the nodes in the path to V_j^S until no such nodes exist as $V_k^S \rightarrow V_j^S$ on the path $s \rightarrow V_j^S$. If there is a subset of nodes on a path, which is a node set on another path, delete the path and then fusion junction as one node, that means the path already exist and not need give a new one.

(4) Calculate the shortest path to V_j^S on the global path:

$$\bar{S}(s, V_j^S) = \min(\sum_{k=1,j=1} (S(s, V_j^S)\Big|Q(V_k^S, V_j^S)))$$

Step 5: Select next node of review text V_j^S and go to Step 1 until all review text nodes to be addressed.

Step 6: Select and set the tightness classification threshold function $f(\varepsilon)$, and proceed as follows:

For all nodes V_j^S, do while not end of j:if there is $\bar{S}(s, V_k^S) \leq f(\varepsilon)$ then all the nodes on the path form a valid path, and correspond a semantic classification, denoted as V_i^C;

Step 7: Output V_i^C and all the nodes on the corresponding path.

We obtain different semantics and its orientations about the reflection of review texts using the Algorithms 1 and 2. The semantics obtained is the meaning mined from one review text or a number of review texts. By analyzing the semantics, we can understand the review texts and grasp the dynamics of people public opinions.

5 Experiment and Discussion

5.1 Dataset

Obviously a collection of internationally recognized data is more supportive to be used to test the effect of our proposed approach. The dataset we use is from Amazon.com collected by Stanford University [19, 20], whose characteristics are shown in Table 1.

Table 1. Review dataset of Amazon

Category	Reviews	Items
Books	22.5M	2.37M
Electronics	7.82M	498K
Sports and outdoors	1.32M	532K
Video games	3.26M	51K
Baby	915K	71.3K

The data set contains 24 categories, about 83.06 million reviews and 9.4 million Items. Each review is composed of *reviewerID* representing the ID of the reviewer, *reviewerName* representing the name of the reviewer, *helpful* representing helpfulness rating of the review, *reviewText* representing the text of the review, *overall* representing the rating of the product, *summary* representing the summary of the review, and the *time* of the review. Each Item is composed of ID of the product, *title* representing the name of the product, *price* representing the price in US dollars (at the time of crawl), *imUrl* representing the url of the product image, as well as *related* representing the related products (i.e. *also bought*, *also viewed*, *bought together*, *buy after viewing*).

In order to facilitate the experiment, we did some preliminary processing for raw data. In this experimental, the data set is divided into two parts: training data and test data. We use the Reviews data as the training set and as a test set the Item with corresponding ID of the product to the reviews. We also randomly select 100K Review data as the test data. Another test set is the 1M samples data obtained from the Review data collection covering the following five datasets, Books, Electronics, Sports and Outdoors, Video Games, Baby. Each corresponding data are also extracted from the Item data set according to the ID of the product in the Review data set. The *related* content of *also bought*, *bought together*, *buy after viewing* of related products indicates that a user purchases another commodity when he have bought or viewed some kind of similar goods. This user behavior reflects that these users have a similar tendency. From the views of semantic analysis, the user who has bought the same goods tends to share similar semantic in their comments. Therefore, it is feasible to maintain the accuracy of user's semantic classification of reviews by considering whether the users have purchased the same items.

5.2 Experimental Results and Analysis

Experiment 1 is to analyze the number of semantic classification under different path lengths. Path length is controlled by the threshold function *f(ε),* see the Algorithm 2. The higher the value of ε is, the longer the path and there are more nodes contained in the same type of classification. This is reflected that the number of semantic classifications is small in the overall number. Conversely, if the value ε is smaller one the semantic relationship between the nodes is closer and there may be less number of nodes to be included in a semantic classification. The number of total classification of semantic is increased. The comparison of the five semantic classifications is shown in Fig. 1.

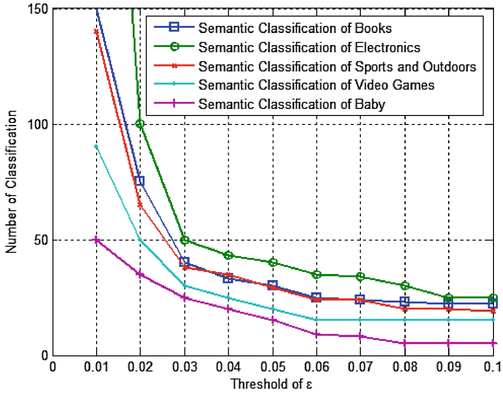


Fig. 2. Comparisons of five semantic classifications under different thresholds

As shown in Fig. 1, it can be observed that when the value of ε is small, the number of semantic classification is larger. The number of classification will increase rapidly when the threshold is less than 0.01. The number of semantic classification will close to a certain value when the threshold value is a larger one. Theoretically, when ε is close to zero, the number of semantic classification is close to the number of reviews of all the users. When it is close to 1, the number of semantic classification will be close to 1. Under the same situation of ε, the number of semantic classification of electronics is large indicates that there are more divergent of user reviews of this kind of commodities. The semantic classifications of the books and sports classes have close numbers whereas the type of baby has smaller number of semantic classifications. It indicates that these users of the baby type have a consensus in opinion.

Experiment 2 is a comparison of the number of semantic classification under different training samples. As we can see from Fig. 3, the number of training samples has a greater impact on the number of semantic classification. When there is a small sample size, the number of semantic classification is lower. When the sample size is increased, the number of semantic categories also increased. The Increasing of the number of semantic classification of will be flattening gently when the number of samples reaches a certain level. Wherein there is a more of electronics semantic types and the types of books and sports are in the intermediate level, there is less semantic type in baby type.

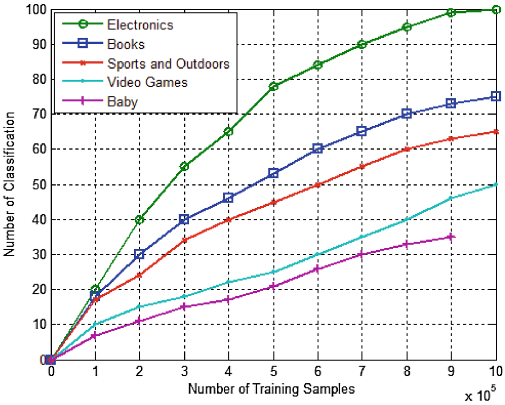


Fig. 3. Comparison of the number of semantic classification under a different number of training samples

The accuracy of the model using for semantic classification is analyzed by the test sample data set. The analysis method uses the proportion of users who buy the same number of items which is belong to the same semantic classification of users, and the proportion of users who purchased the same goods but not in the same semantic classification of users to verify the accuracy. The proportion of user who bought the same product at the same time in the same classification represents an accurate ratio. The proportion of users who buy the same product but not in the same semantic types of users represents the error rate. The comparison of the experimental results is shown in Fig. 2.

Acknowledgments. This paper is a part of research work in the Dalarna University in Sweden. This work was supported by the Natural Science Foundation of Zhejiang (LY16G020012), Major Research Projects of Humanities and Social Sciences in Colleges and Universities of Zhejiang (2014GH015), Science and Technology Innovation Team of Ningbo (2013B82009), Social Development Projects of Ningbo (2012C50045), Research Project of Humanities and Social Sciences of The Ministry of Education (14YJC630210), Public Technology Research and Application Project of Zhejiang (No. 2015C33065).

References

1. Liu, B.: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer, Berlin (2008)
2. Liu, B.: Sentiment analysis and subjectivity. In: Indurkhya, N., Damerau, F.J. (eds.) Handbook of Natural Language Processing, 2nd edn. Chapman & Hall, London (2010)
3. Cambria, E., Grassi, M., Hussain, A., Havasi, C.: Sentic computing for social media marketing. Multimedia Tools Appl. **59**(2), 557–577 (2012)
4. Liu, B.: Sentiment analysis and opinion mining. Synth. Lect. Hum. Lang. Technol. **5**(1), 1–167 (2012). doi:10.2200/S00416ED1V01Y201204HLT016
5. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: HLT 2005 Proceedings of Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, pp. 347–354, October 2005
6. Hobbs, J.R., Riloff, E.: Information extraction. In: Handbook of Natural Language Processing, pp. 1–31 (2010)
7. Bunescu, R.C., Mooney, R.J.: Subsequence kernels for relation extraction. In: Advances in Neural Information Processing Systems, pp. 171–178 (2005)
8. Sarawagi, S.: Information extraction. Found. Trends Databases **1**(3), 261–377 (2008)
9. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project report, Stanford (2009)
10. Kennedy, A., Inkpen, D.: Sentiment classification of movie reviews using contextual valence shifters. Comput. Intell. **22**(2), 110–125 (2006)
11. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177. ACM Press, New York (2004)
12. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retr. **2**(1–2), 1–135 (2008)
13. Turney, P.D.: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of ACL-2002, 40th Annual Meeting of Association for Computational Linguistics, USA, pp. 417–424 (2002)
14. Venugopalan, M., Gupta, D.: Exploring sentiment analysis on twitter data. In: 2015 Eighth International Conference on Contemporary Computing (IC3), Noida, pp. 241–247, 20–22 August 2015
15. Li, G., Liu, F.: A clustering-based approach on sentiment analysis. In: 2010 International Conference on Intelligent Systems and Knowledge Engineering (ISKE), Hangzhou, pp. 331–337, 15–16 November 2010
16. Liu, R., Xiong, R., Song, L.: A sentiment classification method for Chinese document. In: 2010 5th International Conference on Computer Science and Education (ICCSE), Hefei, 24–27 August 2010, pp. 918–922 (2010)
17. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: Proceedings of 7th Conference on International Language Resources and Evaluation (LREC 2010), pp. 1320–1326, May 2010
18. Kao, H.Y., Lin, Z.Y.: A categorized sentiment analysis of chinese reviews by mining dependency in product features and opinions from blogs. In: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), Toronto, ON, vol. 1, 456–459, 31 August – 3 September 2010

19. McAuley, J., Targett, C., Shi, J., van den Hengel, A.: Image-based recommendations on styles and substitutes. In: SIGIR 2015 Proceedings of 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 43–52. ACM, New York (2015)
20. McAuley, J., Pandey, R., Leskovec, J.: Inferring networks of substitutable and complementary products. In: KDD 2015 Proceedings of 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM. New York (2015)

Data Warehouse Quality Assessment Using Contexts

Flavia Serra(✉) and Adriana Marotta

Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay
{fserra, amarotta}@fing.edu.uy

Abstract. Data Warehousing Systems (DWS) are of great relevance for supporting decision making and data analysis. This has been proven over time, through the generalization of its development and use in all kind of organizations. Many researchers have presented the need to incorporate and maintain Data Quality (DQ) in DWS. However, there is no consensus in the research community on how or whether it is possible to define a set of quality dimensions for DWS, since such set may depend on the purpose for which the data are used. Moreover, quality requirements may vary among different domains and among different users. The contribution of this paper is twofold: a study of existing proposals that relate DQ with DWS and with contexts, and a proposal of a framework for assessing DQ in DWS. This proposal is the starting point of a broader and deeper investigation that will allow quality management in DWS.

Keywords: Data quality · Data warehousing system · Data warehouse · Context

1 Introduction

Data Quality (DQ) evaluation in Data Warehouse Systems (DWS) is a very important issue, considering that the main goal of these systems is to give support to decision making. However, despite the existing efforts towards its solution, there are still many open aspects [1–3]. We believe that when considering DQ at least two objectives should be achieved by the system: providing to the final user DQ information about the data he obtains, and controlling and improving DQ throughout its data transformation and loading process.

DWS typically have various components with different goals and characteristics: data sources, ETL (Extraction, Transformation and Load) components, Data Warehouse (DW), data marts (DM), front-end components. Data in each of these components are in different states, regarding granularity, quality, structure, etc. We think that in order to effectively manage DQ in DWS, we must address the problem differently in each DWS component, taking into account the particularities of data and related processes in each one.

The importance and influence of data context in DQ has been stated many years ago [4], and is widely accepted. Also the *fitness for use* approach [5], where it is considered that DQ completely depends on the fitness of the data to the use they will have, is adopted by most of DQ researchers.

W. Cellary et al. (Eds.): WISE 2016, Part II, LNCS 10042, pp. 436–448, 2016.
DOI: 10.1007/978-3-319-48743-4_36

The approach of this work is to assess DQ in DWS by means of considering each system component and the corresponding data context in each one. For example, the context of data in the DW component may be given by associated data from inside and outside the DW, while the context of data in the front-end component may be given by the profile of its user.

In this paper we propose a framework for assessing DQ in DWS using contexts, which is implemented through Datalog rules, and we present an example of its application. Our proposal is based on a thorough study of the existing literature relating DQ, DW and Contexts. Some of the results of this study are also presented in this paper. The contribution of this paper is twofold: (i) a study of existing proposals that relate DQ, DWS and contexts, and (ii) a proposal of a framework for assessing DQ in DWS.

The rest of the document is organized as follows: in Sect. 2 we present some preliminary concepts regarding DQ, DW and Contexts, in Sect. 3 we present the existing work that relate these areas, in Sect. 4 we present our proposal for assessing DQ in DWS, in Sect. 5 we present an example using the proposal framework and finally, in Sect. 6 we present some conclusions and future work.

2 Preliminaries

For the research community DQ is a multi-faceted concept, which is represented by DQ dimensions that address different aspects of data. At the same time, the approach of DQ as *fitness for use* has been widely adopted by this community. A huge set of DQ dimensions have been defined in the literature in the last 20 years, existing a subset that is used by most of the authors, with similar notions and/or definitions. Some works that gather definitions of DQ and DQ dimensions [6, 7], show that there is not an standard or agreement about the set of dimensions that characterize DQ. Recently, in [3] a new effort for organizing DQ dimensions was presented, where six clusters are proposed, which intend to cover the main dimensions: *accuracy*, *completeness*, *redundancy*, *readability*, *accessibility*, *consistency*, *usefulness* and *trust*.

In this work we consider a DQ metamodel for managing DQ, which consists of: *DQ dimensions*, which represent general quality aspects, *DQ factors*, which are more specific quality characteristics that are grouped in each dimension, and *DQ metrics*, which define the way the DQ factors are measured. This way, a DQ dimension may have many corresponding DQ factors and a DQ factor may have many corresponding DQ metrics. For example, for the DQ dimension *accuracy* we can consider the DQ factors *syntactic accuracy* and *semantic accuracy*. Meanwhile, for the DQ factor *syntactic accuracy*, we can consider a DQ metric that calculate it by searching the data value in a dictionary, and another one that applies a rule that determines if the data value has a valid format.

A DW is a database whose data are the result of the extraction, integration, cleaning and diverse transformations of heterogeneous source data, with the goal of giving support to decision-oriented analysis. The most commonly used model for this kind of data is the Multidimensional Model [8], whose intention is to give the analyst a natural way for manipulating the subjects and indicators of the analysis. Data in this model are presented in a n-dimensional space, called *data cube*, where the axis are the *dimensions* of analysis and the points in the space contains the indicators, called *measures*. Each coordinate of the cube is called a *fact*. The dimensions are structured in *hierarchies* that give the criteria for data aggregation, which is an essential operation in multidimensional analysis. Besides, the hierarchies are composed by *levels*, and the instances of a level are called *members*. For example, in the dimension *Time* there may be a hierarchy composed by levels *date*, *month*, *quarter*, *year*. “Q1”, “Q2”, “Q3”, “Q4” are members of the level *quarter*. Figure 1 shows an example of a data cube [8].

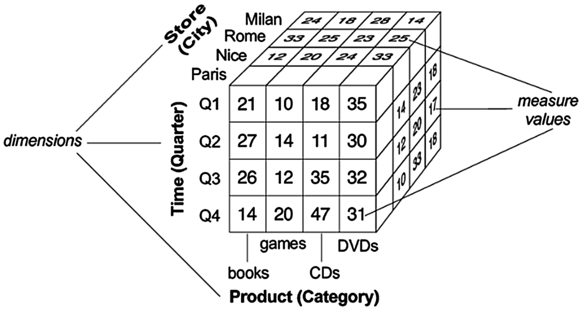


Fig. 1. A three-dimensional cube for sales data having dimensions store, time and product, and a measure amount.

A DWS is an information system whose main component is a DW, and whose general architecture is composed by different components that make possible the whole process from the data sources towards the end-user. In [9] different architectures are presented, but the two-tired architecture, shown in Fig. 2, is stated as the most referenced one. It remarks the separation between the sources and the DW, although it represents the four main stages in the data flow: source layer, data staging, DW layer, and analysis. Moreover, the metadata repository is used in all DWS lifecycle; there

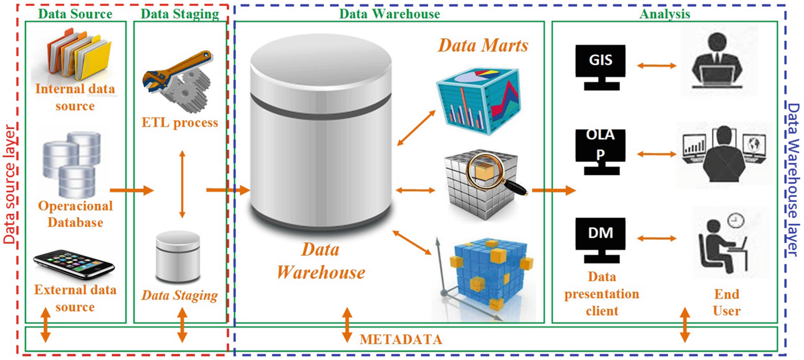


Fig. 2. Two-tier architecture in data warehouse.

exists information about the sources, DM schemes, results of the data quality assessment, users data, etc.

As said before, the importance of considering the context in DQ management is widely recognized. However, it is not possible to find in the literature a concise and globally accepted definition for the concept of context. In fact, there are a lot of conceptualizations and definitions for it, whose approaches depend on the research domain where it is applied. For example, in [10], the context is defined as the possibility of selecting data according to the user environment, while in [11] the authors consider that the context is a set of variables of interest that influence the actions of an agent. In [12] a set of definitions extracted from the Web are analyzed. The authors consider that is difficult to find a relevant definition that satisfies all disciplines and they mention that there are still few ideas about the relevant properties that should be considered when modeling context.

In this work we are interested in *data context*, and the notion of context we use, which is based on the literature, can be synthesized as: the relevant information that influences data content, its interpretation, and the actions over it.

3 Existing Work

In this Section we comment the most important conclusions obtained from a literature review of the existing work that relates the three research topics Contexts, DQ and DW. Afterwards, we present an analysis of these works, putting the focus on DQ tasks and DQ dimensions.

3.1 Connecting Three Research Areas: Context, DQ and DW

The used approach for analysing the research advances in *the use of Contexts in DQ evaluation in DWS* was to also study the proposals that address the combinations of two of the involved topics. Therefore, we present an analysis of the works that focus on DQ-DW, Context-DW and Context-DQ, and then an analysis of the works that address the three topics (Context-DQ-DW).

Data Quality and Data Warehouse (DQ-DW). In spite of the existing consensus on the importance of incorporating and maintaining DQ in DWS, there is not an agreement in the literature on how to do it. Neither is identified the best moment or component in the construction of the DWS, for performing DQ management. Most of the works focus on data cleaning during ETL process, such as the ones presented in [13, 14], and on solving DQ problems in the sources selection stage [15]. The task of evaluating DQ throughout the whole DWS lifecycle is in general ignored or only mentioned but not addressed. In [1] the authors consider that DQ problems in multidimensional repositories still need to be correctly ordered.

In [16] the authors emphasize on the subjectivity of DQ in DWS, considering two aspects: (i) DQ problems may be relevant or not, depending on the decisions to be made, and (ii) data analysts may have different notions and expectations about DQ.

Finally, although many works associate DQ dimensions to DWS [2], an adequate set of DQ dimensions for these systems has not yet been identified and agreed. This fact leads us to question if it is possible or not to define a unique set of DQ dimensions for DWS.

Contexts and Data Warehouse (Context-DW). Some authors consider that the context in a DW is defined by its dimensions, since they are the ones that gives meaning and allow the analysis of the DW measures [17, 18]. Other authors consider that the context is defined by documents content [19, 20], being very frequent that non-structured data obtained from sources as the enterprise intranet, the web or emails, have a relationship with the entities and relationships stored in the DW. They consider that in the data analysis stage the context of the DW facts is also described in documents [20]. On the other hand, they mention that the OLAP (On-line Analytical Processing) performed by the user determines the context of the decision making. The authors claim that although contextual information should be consider when exploiting a DW, there has been very little research in the integration of context in DWS.

Contexts and Data Quality (Context-DQ). Many authors consider that the concept *fitness for use* means that DQ depends on the context where data are used [2, 13, 17, 21–23]. In [23, 24] they consider the context as the data users and their tasks at hand. Besides, in [24] the authors propose to manage DQ considering the construction of information as a process, so DQ is a dynamic measure. In this case, during the information construction, from data to knowledge, DQ would go through different contexts. Additionally, in [23] DQ and information quality are considered different. In particular, for the authors, information quality metrics are necessarily context-dependent while DQ metrics may be absolute.

Many works found in the literature are based on the DQ dimensions classification given in [4], where DQ dimensions are classified in intrinsic, contextual, representational and accessibility, for example [22]. In this work the authors claim that due to the dependence of DQ on the context, despite the wide discussions about DQ dimensions existing in the literature, it does not exist a unified set of DQ dimensions. In [21] the authors remark that most of existing DQ approaches are context-dependent, however the contextual dimension is in general not represented.

Contexts, Data Quality and Data Warehouse (Context-DQ-DW). Few works relating the three research areas are found in the literature. The proposal presented in [2] integrates DQ during the whole DW development process, in particular in the requirements analysis phase. The authors mention that some DQ dimensions are objective and others are subjective, and the subjectivity is given by the users by their DQ requirements.

Meanwhile, in [17] contexts are represented through dimensions that include hierarchies. According to the authors, DQ cannot be evaluated without a contextual knowledge about the production and use of data.

Finally, in [25] the authors remark that DQ is context-sensitive by nature, and therefore it must be evaluated in the context of the business where data will be used.

They also state that research in DQ evaluation has only focused on the identification of DQ dimensions and factors, and in particular for decision support systems relevant DQ factors has not been identified and context has not been considered.

3.2 Analysis from DQ Perspective

In this Section, the reviewed works are analyzed with respect to two aspects: first, the DQ tasks presented and the consideration of the context for the proposal, and second, the DQ dimensions used or defined in the proposal.

DQ Tasks. Different DQ tasks are addressed in the considered literature: data profiling, data cleaning and data evaluation, and only some of the works propose the use of context when performing the task. Table 1 shows the references of the works that address each task and if they consider or not the context for their proposal.

Table 1. Data quality tasks

Task	Takes into account the context	
	Yes	No
Analysis	[23, 26]	[15, 21, 22, 24, 25, 27, 28]
Data cleaning	[29]	[13, 14, 17, 26, 27, 30]
Measurement	[23, 25, 26, 29, 31]	[15, 21, 22, 24, 27, 28]

As can be observed, DQ Measurement tasks are the ones that most use context in the analyzed proposals. The authors of these works remark the importance of context consideration in the DQ metrics definitions, and show how context elements such as the user task, can modify the DQ measurement results.

Quality Dimensions. Another important result obtained from the literature review, is the identification of the DQ dimensions proposed in the different works. We group the papers according to their topics, in Context-DQ and DQ-DW, and for each group we present the proposed DQ dimensions. Table 2 show the most relevant DQ dimensions (for space reasons, 13 from 54) proposed in Context-DQ papers, while Table 3 shows the most relevant DQ dimensions (for space reasons 10 from 21) presented in DQ-DW papers.

It is worth noting the variety of terminology and concepts for DQ dimensions that can be found in the bibliography, for example *accuracy* and *correctness*, which refer to the same concepts. This confirms the lack of standardization of the concepts in DQ research area. As can be seen, much more DQ dimensions have been found in the works of DQ and context than in the ones of DQ and DW. In both cases the dimensions *accuracy*, *completeness* and *timeliness* are the most referred.

Table 2. Quality dimensions in articles focused on DQ and CTX.

Dimension	Article
Accessibility	[22]
Accuracy	[22, 28, 31]
Completeness	[21, 22, 28, 31]
Consistency	[21, 28]
Correctness	[24]
Freshness	[31]
Granularity	[25]
Precision	[31]
Relevancy	[22, 25]
Security	[22, 31]
Timeliness	[21–23, 28]
Traceability	[22]
Usefulness	[24]

Table 3. Quality dimensions in articles focused on DQ and DW.

Dimension	Article
Accuracy	[1, 27, 30]
Completeness	[1, 27, 30, 32]
Consistency	[1, 27, 32]
Correctness	[32]
Reasonableness	[15]
Temporality	[15]
Timeliness	[1, 15, 30]
Transparency	[32]
Trust-worthiness	[15, 32]
Uniqueness	[1]

4 Framework for DQ Assessment in DWS

According to the articles analyzed before, we develop a proposal whose purpose is to define a framework that gives support to DQ assessment in DWS. For this, we consider the different components that make up the SDW (see Fig. 2) and the different contexts that have influence on such components. One purpose of this work is to define contexts, which may influence DQ assessment, along the entire lifecycle of the DWS, considering the different contexts that data go across from the DW until they are used by the end-users [24]. Our proposal does not focus on data sources and ETL layers quality issues, since many researches have already addressed them (especially data cleaning inclusion in ETL).

4.1 Context in Each DW Component

In this section we present and define contexts for the components in the DW layer. Each component with its context is presented in Fig. 3. In the following we describe the elements that determine each context.

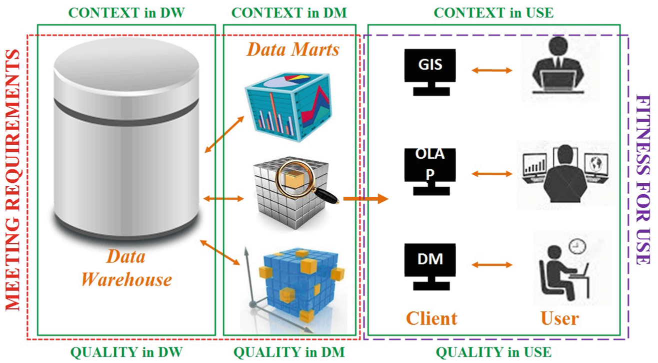


Fig. 3. Two quality approaches in a DWS to evaluate data quality according to the context.

Context in the Data Warehouse (DWC): It is defined by data in the DW, documents, e-mails and other data external to the DW. All these elements are related to data stored in the DW.

Context in the Data Mart (DMC): A Data Mart contains a subset of the data stored in the DW, which had been transformed, and is directed to a specific analysis domain (e.g. a section in the organization). Hence, for us, the DMC is determined by a set of rules that describe properties, constraints and quality requirements specific to the corresponding analysis domain.

Context in Use (CiU): The CiU is the context in the data presentation layer, and is determined by data that describe the end-user. These data can be geographical location, language, role, requirements (of data or quality), etc. The context could be one of them or a combination of them. For example, the DQ requirements could be a minimum level of data accuracy or data completeness.

4.2 Data Quality According to the Context

For the quality assessment in the DW components, taking into account the contexts introduced before, this work is supported by two quality approaches: *Crosby's Meeting Requirements* (compliance with the requirements) [33] and *Juran's Fitness for Use* (meeting the needs of the user) [5]. The former is applied for DQ assessment in two components, DW and DM, while the latter is applied for DQ assessment in the data presentation layer (shown in Fig. 3). Based on these quality approaches DQ is defined according to the context in each DW component:

Quality in the Data Warehouse (DWQ) depends on the DWC, therefore the DQ metrics for the DW are defined using this context.

Quality in the Data Mart (DMQ) depends on the DMC, therefore the DQ metrics for the DM are defined using this context.

Quality in Use (QiU) depends on the CiU, therefore the DQ metrics for the presentation layer are defined using this context.

The concepts we have just defined constitute the base to apply the following steps for defining the DQ metrics for assessing DQ in DWS:

1 – Select the **component** to be assessed: DW, DM or Presentation
2 – If the component is DM then specify **Domain** and **Domain Rule(s)** Else
If the component is Presentation, specify **User data**
3 – Determine **DQ dimension**
4 – Determine **DQ factor**
5 – Define **DQ metric**: Name, contextualizing object, contextualized object, granularity, description and result type.

4.3 Implementation Using Datalog

We implement the context-based DQ metrics using Datalog, since it allows us to represent the DWS data, the defined contexts and the metrics as a set of logical rules, which can also be executed performing the DQ measurements. Our model is based on the model of [34], and the example used for applying it, is a supermarket chain called "BigSales" that maintains information about its sales in a DWS. Figure 4 shows the conceptual multidimensional model (following MultiDim model [8]), where the DW dimensions (Products, Time and Store) with their hierarchies and levels, and the DW dimensional relationship (Sales) with its measure (amount-of-sales), are shown.

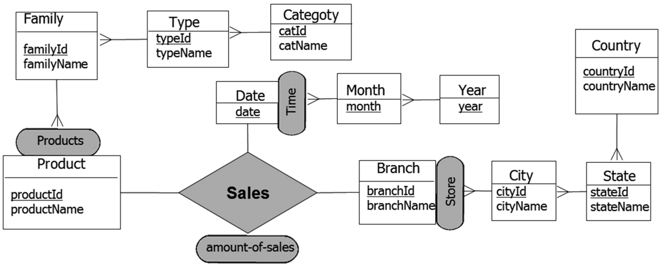


Fig. 4. Hierarchies for each dimension in the DW and the dimensional relationship "Sales".

According to the model of [34], facts are represented through abstract entities, e.g. *AFactQty*(Sales, s1,50), where s1 is a fact identifier and 50 is its measure value. The *aggr* predicate associates an abstract fact with a level member, e.g. *aggr*(X, Store, branchId, 31) means that the dimension is Store, the level is branch and the member has branchId 31, for the abstract fact X. The rules represent the rollup operations between level members and for our example they are of the form:

aggr(X, Products, familyId, chocolate) :- aggr(X, Product, productID, chocolate bars)

The rollup operations between the level members productId and familyId

aggr(X, Store, cityId, MVD) :- aggr(X, Store, branchId, 30)

The rollup operations between the level members branchId and cityId

aggr(X, Time, month, 5 − 2013) :- aggr(X, Time, date, 30−5−2013)

The rollup operations between the level members date and month

idToName(MVD, Montevideo). Given an id returns the corresponding name

AFactQty(Sales, s1, 50). s1 is a fact identifier and 50 is its measure value

5 Using the Framework

Due to lack of space, we only present one case, out of six developed, that illustrates how the proposed framework is used.

Metric Definition

Component: DM **Domain:** Sales

Domain Rule (R_{Sales}): "The branch's name structure must be p_1-p_2-p_3, where p_1 is the supermarket's name, p_2 is the city's name to which it belongs (cityName in the city level) and p_3 is the branch's identifier (branchId in the branch level)"

Quality dimension: Accuracy **Quality factor:** Syntactic accuracy

Quality metric: *dmq_Example* **Contextualizing object:** R_{Sales}

Contextualized object: branch level members of Store dimension

Granularity: Attribute

Description: For each level member (with branchName b) of the branch level, the metric verifies that b has the structure p_1-p_2-p_3 where p_1 must be "BigSales" (Supermarket's name), p_2 must be the value in cityName and p_3 the value in brancheId.

Result: 1 (if the attribute value verifies R_{Sales}), 0 otherwise.

Metric Implementation

The *Contextualizing object* is the domain rule R_{Sales}, and the rules *aggr*(X, Store, branchId, B), *aggr*(X, Store, cityId, A) and *idToName*(A, C) are used to represent it. The *dmq_Example* DQ metric uses this context and other rules.

context(X,B,C):-aggr(X,Store,branchId,B),aggr(X,Store,cityId,A),idToName(A, C).

Where *A* is the id of a city, *B* is the id of a branch and *C* is the city's name for *A*. For each abstract fact X, the *context* predicate returns the id of the branch and the city's name for this branch.

dmq_Example(X, N):- context(X,B,C), aggr(X,store,branchId,S), idToName(S,N),
branchStructName1(S,Y), branchStructName2(S,Z), branchStructName3(S,W),
'BigSales' = Y, B = Z, C = W.

Where *S* is the branchId, *N* is the branchName in the DW for the brancheId, *Y* is the real value of p_1 in the branchName's structure (p_1-p_2-p_3), *Z* is the real value of p_2, *W* is the real value of p_3. For each abstract fact X, the *dmq_Example* DQ metric returns the names of all branches that meet the rule R_{Sales} (which correspond to the result = 1 specified in the metric definition)

6 Conclusion and Future Work

In this work we presented a literature review that showed the importance of considering the context when assessing DQ, in particular in DWS. Based on the performed literature analysis and some obtained results, a framework for DQ assessment in DWS was presented. This framework allows and leads the user to the consideration of the contextual nature of data quality, which was widely analyzed in the literature, however, we have not yet found a research solving this issue in DWS.

The proposed framework is based on the definition of contexts for the different DWS components, and DQ metrics that use these contexts. It contains a set of steps that must be applied for defining the context-based DQ metrics. The main advantage of this framework is that it helps the DQ expert to identify the context that influences DQ in each DWS component, and to define appropriate context-based DQ metrics.

An implementation using Datalog is presented too, which allows the representation of the metrics and their execution. Finally, an example is shown as a proof of concept.

As ongoing work we are implementing a case study with real data, where the framework is applied, and as future work we are planning to formalize the proposed models and develop a complete framework that allows DWS DQ assessment.

References

1. Gongora de Almeida, W., de Sousa, R.T., de Deus, F.E., Amvame Nze, G.D., Lopes de Mendonca, F.L.: Taxonomy of data quality problems in multidimensional data warehouse models. In: 8th Iberian Conference on Information Systems and Technologies (CISTI), pp. 1–7 (2013)
2. Munawar, M., Salim, N., Ibrahim, R.: Towards data quality into the data warehouse development. In: IEEE Ninth International Conference on DASC, pp. 1199–1206 (2011)
3. Batini, C., Scannapieco, M.: Data and Information Quality – Dimensions, Principles and Techniques. Data-Centric Systems and Applications. Data-Centric Systems and Applications. Springer, Heidelberg (2016). ISBN 978-3-319-24104-3
4. Strong, D.M., Lee, Y.W., Wang, R.Y.: Data quality in context. Commun. ACM **40**(5), 103–110 (1997)
5. Juran, J., Godfrey, A.B.: Quality Handbook. Republished McGraw-Hill, New York (1999)
6. Batini, C., Scannapieco, M.: Data Quality: Concepts Methodologies and Techniques. Springer, Heidelberg (2006)
7. Scannapieco, M., Catarci, T.: Data quality under a computer science perspective. Arch. Comput. **2**, 1–15 (2002)

8. Malinowski, E., Zimányi, E.: Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications. Data-Centric Systems and Applications. Springer, Heidelberg (2008)
9. Golfarelli, M., Rizzi, S.: Data Warehouse Design: Modern Principles and Methodologies. McGraw-Hill Inc., New York (2009)
10. Ciaccia, P., Torlone, R.: Modeling the propagation of user preferences. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 304–317. Springer, Heidelberg (2011). doi:10.1007/978-3-642-24606-7_23
11. Bolchini, C., Curino, C.A., Orsi, G., Quintarelli, E., Rossato, R., Schreiber, F.A., Tanca, L.: And what can context do for data? Commun. ACM **52**(11), 136–140 (2009)
12. Bazire, M., Brézillon, P.: Understanding context before using it. In: Dey, A., Kokinov, B., Leake, D., Turner, R. (eds.) CONTEXT 2005. LNCS, vol. 3554, pp. 29–40. Springer, Heidelberg (2005)
13. Ali, K., Warraich, M.A.: A framework to implement data cleaning in enterprise data warehouse for robust data quality. In: ICIET International Conference, pp. 1–6 (2010)
14. Santos, V., Belo, O.: Modeling ETL data quality enforcement tasks using relational algebra operators. Procedia Technol. **9**, 442–450 (2013). CENTERIS 2013
15. Prat, N., Madnick, S.: Measuring data believability: a provenance approach. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences, pp. 393–393 (2008)
16. Daniel, F., Casati, F., Palpanas, T., Chayka, O.: Managing data quality in business intelligence applications. In: QDB/MUD, pp. 133–143 (2008)
17. Hamad, M.M., Jihad, A.A.: An enhanced technique to clean data in the data warehouse. In: Developments in E-systems Engineering (DeSE), pp. 306–311 (2011)
18. Silva Souza, V.E., Mazón, J.N., Garrigós, I., Trujillo, J., Mylopoulos, J.: Monitoring strategic goals in DW with awareness requirements. In: SAC 2012, pp. 1075–1082. ACM, New York (2012)
19. Thollot, R., Brauer, F., Barczynski, W.M., Aufaure, M.A.: Text-to-query: dynamically building structured analytics to illustrate textual content. In: EDBT 2010, pp. 14:1–14:8. ACM (2010)
20. Perez, J.M., Berlanga, R., Aramburu, M.J., Pedersen, T.B.: Towards a data warehouse contextualized with web opinions. In: IEEE International Conference on ICEBE 2008, pp. 697–702 (2008)
21. Helfert, M., Foley, O.: A context aware information quality framework. In: Fourth International Conference on COINFO 2009, pp. 187–193 (2009)
22. Moges, H.T., Dejaeger, K., Lemahieu, W., Baesens, B.: A multidimensional analysis of data quality for credit risk management: new insights and challenges. Inf. Man **50**(1), 43–58 (2013)
23. Alberts, D.S., Vassiliou, M., Agre, J.: C2 information quality: an enterprise systems perspective. In: MILCOM 2012, pp. 1–7 (2012)
24. McNab, A.L., Ladd, D.A.: Information quality: the importance of context and trade-offs. In: 47th Hawaii International Conference on HICSS, pp. 3525–3532 (2014)
25. Sundararaman, A.: A framework for linking data quality to business objectives in decision support systems. In: 3rd International Conference on TISC 2011, pp. 177–181 (2011)
26. Milani, M., Bertossi, L., Ariyan, S.: Extending contexts with ontologies for multidimensional data quality assessment. In: IEEE 30th International Conference on ICDEW 2014, pp. 242–247 (2014)
27. Sidi, F., Ramli, A., Jabar, M.A., Affendey, L.S., Mustapha, A., Ibrahim, H.: Data quality comparative model for data warehouse. In: International Conference on CAMP 2012, pp. 268–272 (2012)

28. Hazen, B.T., Boone, C.A., Ezell, J.D., Jones-Farmer, L.A.: Data quality for data science, predictive analytics, and big data in supply chain management: an introduction to the problem and suggestions for research and applications. Int. J. Prod. Econ. **154**, 72–80 (2014)
29. Dasu, T., Loh, J.M., Srivastava, D.: Empirical glitch explanations. In: KDD 2014, pp. 572–581. ACM, New York (2014)
30. Huang, Z., Peng, H.: Improving uncertain data-quality through effective use of knowledge base. In: 4th International Conference on WiCOM 2008. pp. 1–4 (2008)
31. Zheng, D., Wang, J., Kerong, B.: Evaluation of quality measure factors for the middleware based context-aware applications. In: 11th IEEE/ACIS, ICIS 2012, pp. 403–408 (2012)
32. Wieder, B., Ossimitz, M.L.: The impact of business intelligence on the quality of decision making a mediation model. Procedia Comput. Sci. **64**, 1163–1171 (2015)
33. Crosby, P.B.: Quality Is Free: The Art of Making Quality Certain. McGraw-Hill, New York (1979)
34. Marotta, A., Vaisman, A.: Rule-based multidimensional data quality assessment using contexts. In: Madria, S., Hara, T. (eds.) DaWaK 2016. LNCS, vol. 9829, pp. 299–313. Springer, Heidelberg (2016). doi:10.1007/978-3-319-43946-4_20

Author Index