Lecture Notes in Social Networks

Özgür Ulusoy
Abdullah Uz Tansel
Erol Arkun *Editors*

Recommendation and Search in Social Networks

Springer

Lecture Notes in Social Networks

More information about this series at http://www.springer.com/series/8768

Özgür Ulusoy · Abdullah Uz Tansel
Erol Arkun
Editors

Recommendation and Search in Social Networks

Springer

Editors
Özgür Ulusoy
Department of Computer Engineering
Bilkent University
Ankara
Turkey

Erol Arkun
Department of Computer Engineering
Bilkent University
Ankara
Turkey

Abdullah Uz Tansel
Department of Statistics and Computer Information Systems
Baruch College, CUNY
New York, NY
USA

ISSN 2190-5428 ISSN 2190-5436 (electronic)
Lecture Notes in Social Networks
ISBN 978-3-319-14378-1 ISBN 978-3-319-14379-8 (eBook)
DOI 10.1007/978-3-319-14379-8

Library of Congress Control Number: 2014959199

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

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

Preface

This book is a timely collection of 12 chapters that present the state of the art in various aspects of social search and recommendation systems. Within the broader context of social network analysis, it focuses on important and upcoming topics of social search and recommendation systems. We believe that the book is a coherent collection of chapters which is not easily accomplished in edited volumes.

Many of the chapters are expanded versions of the best papers presented in the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'2013), which was held in Niagara Falls, Canada in August 2013. The papers were selected based on the reviews for the conference and then were improved substantially by the authors. In addition to the selected papers, the book also features invited chapters in the field of social search and recommendation systems.

The first chapter, by Xinyue Wang, Laurissa Tokarchuk, Felix Cuadrado and Stefan Poslad, presents an adaptive crawling model to detect emerging popular topics, by searching for highly correlated data for the events of interest. In the next chapter, Yuki Urabe, Rafal Rzepka, and Kenji Araki propose an emoticon recommendation system based on users' emotional statements and evaluate its performance in comparison to other such recommendation systems. Then, Georgios Alexandridis, Giorgos Siolas, and Andreas Stafylopatis present a novel random walk social recommendation approach based on rejection sampling.

In "Social Network Derived Credibility," Erica Briscoe, Darren Appling and Heather Hayes explore the use of social network properties as a basis for determining credibility. In the next chapter, Benjamin C.M. Fung, Yan'An Jin, Jiaming Li, and Junqiang Liu propose a method to anonymize the social network with the goals of hiding the identities of the participants and preserving the frequent sharing patterns within a community. In the following chapter, Cheng Chen, Kui Wu, Venkatesh Srinivasan, and Xudong Zhang present a new detection mechanism, using both semantic and nonsemantic analysis, to identify a special group of online users, called hidden paid posters.

In their work, Sogol Naseri, Arash Bahrehmand, and Chen Ding strive to enhance recommendation accuracy through the use of a new similarity metric

which is based on social tagging information. They also present a recommendation method that applies user similarity for finding the most interesting items to target user's taste. Ali Khodaei, Cyrus Shahabi, and Sina Sohangir propose a new model, called persocial relevance model utilizing social signals to improve the web search, in their chapter titled "Personalization of Web Search Using Social Signals". Linhong Zhu, Sheng Gao, Sinno Jialin Pan, Haizhou Li, Dingxiong Deng, and Cyrus Shahabi provide a formulation for the informative sentence selection problem in opinion summarization as a community leader detection problem. Then, they present new algorithms to identify communities and leaders.

The chapter by Hasan Shahid Ferdous, Mashrura Tasnim, Saif Ahmed, and Md. Tanvir Alam Anik explores differences in searching habits of the social networking sites in different regions of the world based on their level of economic development. In "Evolutionary Influence Maximization in Viral Marketing", Sanket Naik and Qi Yu propose a new framework to effectively apply viral marketing in a dynamic social network. The last chapter by Alessia Amelio presents an investigation of the voting behavior of the Italian Parliament by employing methods in data mining and network analysis fields.

We would like to express our appreciation to all contributing authors; without their cooperation this book would not have been possible. Our special thanks are due to Ms. Pauline Lichtveld for her support and to Dr. Tansel Özyer who diligently supported and coordinated the entire process of preparing this timely volume on social network analysis.

Özgür Ulusoy
Abdullah Uz Tansel
Erol Arkun

Contents

Contributors

Saif Ahmed BUET, Dhaka, Bangladesh

Georgios Alexandridis School of Electrical and Computer Engineering, National Technical University of Athens, Zografou, Athens, Greece

Alessia Amelio Institute for High Performance Computing and Networking (ICAR), National Research Council of Italy (CNR), Rende, CS, Italy

Md. Tanvir Alam Anik BUET, Dhaka, Bangladesh

Darren Scott Appling Georgia Tech, Atlanta, GA, USA

Kenji Araki Hokkaido University, Graduate School of Information Science and Technology, Sapporo, Japan

Arash Bahrehmand Department of Information and Communications Technologies, Universitat Pompeu Fabra Barcelona, Barcelona, Spain

Erica J. Briscoe Georgia Tech, Atlanta, GA, USA

Cheng Chen University of Victoria, Victoria, Canada

Felix Cuadrado School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK

Dingxiong Deng University of Southern California, Los Angeles, USA

Chen Ding Department of Computer Science, Ryerson University, Toronto, Canada

Hasan Shahid Ferdous University of Melbourne, Parkville, VIC, Australia

Benjamin C.M. Fung McGill University, Montreal, QC, Canada

Sheng Gao Institute for Infocomm Research, Singapore, Singapore

Heather Hayes Georgia Tech, Atlanta, GA, USA

Yan'an Jin Huazhong University of Science and Technology, Hubei University of Economics, Hubei, China

Ali Khodaei Yahoo! Corporation, Sunnyvale, CA, USA

Jiaming Li IBM Canada Software Lab, Toronto, ON, Canada

Haizhou Li Institute for Infocomm Research, Singapore, Singapore

Junqiang Liu Zhejiang Gongshang University, Zhejiang, China

Sanket Anil Naik Rochester Institute of Technology, Rochester, NY, USA

Sogol Naseri Department of Computer Science, Ryerson University, Toronto, Canada

Sinno Jialin Pan Institute for Infocomm Research, Singapore, Singapore

Clara Pizzuti Institute for High Performance Computing and Networking (ICAR), National Research Council of Italy (CNR), Rende, CS, Italy

Stefan Poslad School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK

Rafal Rzepka Hokkaido University, Graduate School of Information Science and Technology, Sapporo, Japan

Cyrus Shahabi Department of Computer Science, University of Southern California, Los Angeles, CA, USA

Georgios Siolas School of Electrical and Computer Engineering, National Technical University of Athens, Zografou, Athens, Greece

Sina Sohangir GraphDive Company, Menlo Park, CA, USA

Venkatesh Srinivasan University of Victoria, Victoria, Canada

Andreas Stafylopatis School of Electrical and Computer Engineering, National Technical University of Athens, Zografou, Athens, Greece

Mashrura Tasnim BUET, Dhaka, Bangladesh

Laurissa Tokarchuk School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK

Yuki Urabe Hokkaido University, Graduate School of Information Science and Technology, Sapporo, Japan

Xinyue Wang School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK

Kui Wu University of Victoria, Victoria, Canada

Qi Yu Rochester Institute of Technology, Rochester, NY, USA

Xudong Zhang Peking University, Haidian District, Beijing, China

Linhong Zhu Information Sciences Institute, Los Angeles, USA

Adaptive Identification of Hashtags for Real-Time Event Data Collection

Xinyue Wang, Laurissa Tokarchuk, Felix Cuadrado and Stefan Poslad

Abstract The widespread use of microblogging services, such as Twitter, makes them a valuable tool to correlate people's personal opinions about popular public events. Researchers have capitalized on such tools to detect and monitor real-world events based on this public, social, perspective. Most Twitter event analysis approaches rely on event tweets collected through a set of predefined keywords. In this paper, we show that the existing data collection approaches risk losing a significant amount of event-relevant information. We propose a refined adaptive crawling model, to detect emerging popular topics, using hashtags, and monitor them to retrieve greater amounts of highly associated data for the events of interest. The proposed adaptive crawling model expands the queries periodically by analyzing the traffic pattern of hashtags collected from a live Twitter stream. We evaluated this adaptive crawling model with a real-world event. Based on the theoretical analysis, we tuned the parameters and ran three crawlers, including one baseline and two adaptive crawlers, during the 2013 Glastonbury music festival. Our analysis shows that adaptive crawling based on a Refined Keyword Adaptation algorithm outperforms the others. It collects the most comprehensive set of keywords, and with the minimal introduction of noise.

Keywords Twitter · Hashtag · Information adaptation · Information retrieval · Event analysis

X. Wang (✉) · L. Tokarchuk · F. Cuadrado · S. Poslad
School of Electronic Engineering and Computer Science,
Queen Mary University of London, London, UK
e-mail: xinyue.wang@qmul.ac.uk

L. Tokarchuk
e-mail: laurissa.tokarchuk@qmul.ac.uk

F. Cuadrado
e-mail: felix.cuadrado@qmul.ac.uk

S. Poslad
e-mail: stefan.poslad@qmul.ac.uk

Ö. Ulusoy et al. (eds.), *Recommendation and Search in Social Networks*,
Lecture Notes in Social Networks, DOI 10.1007/978-3-319-14379-8_1

1 Introduction

The enormous popularity of Microblogs, combined with their conversational characteristic [1] (leading to multiple short updates and used as a medium to express opinions) has led them to become one of the most popular platforms for researchers to extract public information. Early attempts were conducted to identify characteristics of information diffusion and users' behavior on the entire microblogsphere [7, 12, 15]. Nowadays, the research focus has shifted to more specific problems, such as real-world event detection [11] and event summarization [5].

As one of the most popular microblogging services, Twitter[1] provides people with a platform to share their observations and opinions online. This simple version of a blog service allows users to post short messages (tweets) up to 140 characters. Users can not only update their thoughts through the website, but also post tweets using their mobile devices through either a cellular network or Short Message Service (SMS). This easy access to Twitter facilitated the dramatic growth of the number of Twitter users. With thousands of posts published every second,[2] Twitter also becomes a precious resource pool for researchers to analyze public reaction and behavior under event scenario.

For instance, recent research has examined the use of such tools, primarily Twitter-based, to get knowledge about ongoing affairs [4, 6, 9, 19], or even to dig out hints of upcoming events [2, 8]. Becker et al. use Twitter, along with other social media sites, to retrieve content associated with a planned event [4]. Sakaki et al. use Twitter to detect the occurrence and location of earthquakes even before the disaster hits [2].

In order to identify and analyze events among the entire Twittersphere (also called Twitterverse), a comprehensive dataset describing the event is compulsory. The majority of collection techniques collect tweets from the live Twitter stream by matching a few search keywords or hashtags. For example, Starbird and Palen collected information about the 2011 Egyptian uprising by using the keywords "*egypt, #egypt, #jan25*" [3], Nichols et al. collected sport related tweets using keywords "*worldcup*" and "*wc2010*" [18]. However, the set of predefined keywords is subjective and can easily lead to incomplete data. Moreover, even given expert knowledge, keywords and specialized hashtags often arise in the midst of such events. For example, Fig. 1 shows two tweets relating to the London 2012 Olympics (the football event). It is straightforward to determine that the first one is related to the 2012 Olympics football event, whereas the second one, which refers to the same event, is much harder to distinguish. Figure 2 illustrates how this will result in the loss of a significant amount of event-related information. The blue solid line is the traffic generated by using both *Olympic* or *#teamgb* as keywords, while the red dashed line represents the volume of tweets solely retrieved using a keyword *Olympic*. It is clear that the trend for both lines is the same, but the volume varies. A larger amount of

[1] Twitter Home page, https://twitter.com/.

[2] New Tweets per second record, and how: https://blog.twitter.com/2013/new-tweets-per-second-record-and-how.

Fig. 1 Tweets about the 2012 Olympic games

Goal! Aaron Ramsey. Penalty. GB 1-1 Korea. **#football #olympic**

And just like that **#FIFA** awards **#GBR** a penalty. **#GBRvKOR**

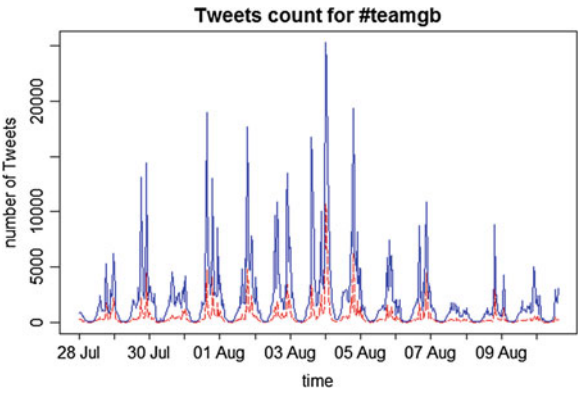

Fig. 2 Comparison of tweet volume crawled by keyword *Olympic* (lower, *red dashed line*) versus *Olympic &#teamgb* (higher, *blue solid line*) during the 2012 London Olympics

event information can be fetched if other keywords are introduced. This issue is even more severe when using Microblogs for situation awareness during emergencies or disasters. People will communicate their observation and perception about events, even without explicitly mentioning the title of the event [18].

Moreover, Twitter's APIs data access restrictions,[3,4,5] greatly complicate collecting all the social media documents corresponding to one event. Lanagan et al. mentioned that incomplete tweet datasets significantly affects the performance of their event detection algorithm [17]. In fact, the Twitter API restrictions not only introduce difficulties on live tweets retrieval, but they also make it harder to recapture data once the events of interest are finished.

In this paper, we aim to present an automatic event content collection method that gathers a set of tweets, without preliminary knowledge of the events, by just relying on initial search terms for live events. We introduce an adaptive microblogging crawling model that allows comprehensive information about an event to be retrieved. By embedding the Keyword Adaptation Algorithm (KwA), this adaptive crawling model can collect an extended set of specific instances of an event. This is achieved by monitoring the Twitter live stream with only the initial keywords, without manual modification of the search terms. In designing the adaptive crawling

[3] Search API only returns tweets within 7 days, and the rate limit of Search API is not specified in the official documentation. Version 1.0.

[4] Streaming API provides real-time services but only returns 1 % of total number of tweets. Version 1.0.

[5] At time of publication, access to the full Firehose stream of tweets is allowed only if a large amount of money is paid, e.g., PowerTrack costs $2,000 per month plus $0.10 per 1,000 tweets delivered. Retrieved from: http://gnip.com/pr_announcing_power_track.

model, the challenge is to identify extra search terms, beyond the original keywords, appearing in content related to the event in question. Specifically, compared with the previous work [29], which only evaluates the KwA algorithm theoretically with existing datasets, the novel contributions of the paper are as follows:

- We investigate the use of trend detection method in our proposed adaptive crawling model and prove that it is insufficient to identify event relevant topics.
- We examine the proposed adaptive crawling model for real-time events by retrieving multiple datasets with an exemplar type of real-time event.
- We demonstrate that the adaptive crawling based on a Refined Keyword Adaptation Algorithm (RKwA) identifies event topics in real-time. Also, its collects additional relevant tweets, while greatly reducing the amount of irrelevant information.

The remainder of this paper is divided into four sections. Section 2 introduces the related work and distinguishes our work from existing work; Sect. 3 introduces the functions and restrictions of Twitter service; Sect. 4 details the proposed adaptive crawling models; Sect. 5 reports the evaluation of our technique, showing its performance over the 2013 Glastonbury Festival; and finally Sect. 6 concludes our work and discusses some future directions.

2 Related Work

Online content collection and analysis has been a popular research issue for years. Much work exists pertaining to structured articles collection from online platforms [16, 21]. Later, researchers tried to improve the traditional content collection and analysis approaches by taking advantage of additional information [30]. Some researches detected latent features (e.g., topics) to obtain a better understanding of the event in question [24]. However, the differences between traditional websites (i.e., news portals and blogs) and Microblogs with respect to resource deployment and contents structure make the transplant of Web-based to microblog methods difficult. This section will review the existing work relating to online crawling, topic detection, and similarity measurement of text, specifically under the Twitterverse.

Crawling a set of online documents, relating to an event of interest, can be achieved by simple keywords searching. This approach has been adopted by some early attempts on tweet collection and analysis [2, 3], but crawling on a predefined keywords set did not provide satisfactory results. In addition to these kind of aforementioned approaches, attempts to use more than keywords as search criteria have also been made [4, 9, 13]. For example, Becker et al. examine the use of precision and recall-oriented strategies to automatically identify event features, then generating queries to retrieve content from diverse social media sites for planned events. Unlike our proposed model, which solely relies on initial terms, they use event announcements from sites such as Last.fm[6] to aid query formulation [4]. Rather

[6] Last.fm website: http://www.last.fm/.

than use other websites, Fabian et al. leverage several metrics from Twitter, such as users' profiles, semantics meanings and metadata of tweets, to generate new search criteria from news websites [9]. Although this additional material facilitates a higher precision and recall rate on search results, the processing cost of these exponentially increases. While planned events can draw on extra material from announcements and news sites, such material for unplanned events is almost impossible to obtain. Furthermore, these solutions were designed to improve the user experience of interactive searching rather than collect additional event-related tweets for real world ongoing affairs. In addition, event tweets can also be fetched from a particular group of target users [31, 32]. This kind of approach chooses the users that are involved in or related to the event as the initial seed for collection. For example, 11 car-related companies are selected as seeds when collecting tweets for 2012 Super Bowl [31]. It is similar to the pre-defined keyword crawler as the initial seeds are fixed. Although our crawling target is different, it is possible to apply our idea in their scenario to support seed adaptation.

Recently, Twitter has attracted unprecedented attention with the research efforts on the detection of trending topics under different circumstances. Some researchers report some success with the detection of event topics and content in large Twitter datasets [2, 8, 17]. However, these types of techniques analyse tweets and track the inherent topic on a large datasets which only represents the state of the Twitterverse at a particular point in time. Namely, these researchers concentrated on building an accurate model in an offline fashion. On the other hand, some researchers explored the traffic characterization of text streams [22, 23] for real time identification of the emerging topics. This kind of approach tracks the evolution of topics by identifying frequent terms in a specific time interval. Rather than identifying general trending topics for multiple events, our objective is to identify an extended set of topic terms for a single event. In addition, the proposed model utilizes the relations between topic terms rather than measuring them separately. The other kind of online topic detection approach builds a model for each topic by capitalizing on the statistical relations between vocabularies [24, 25]. Their conclusion is based on the observation that some particular words appear in the documents belonging to the same topic more frequently, while others less so. However, the main drawback is that they rely on the prior training to construct an accurate topic model. Explicitly, they require the use of human annotated tweets during training stage, i.e. background knowledge about the event need to be known in advance, which is not feasible enough for real-time topic detection. Moreover, statistic based approaches for short text modeling in microblogging environments remain an open research issue since the effectiveness of a trained topic model can be highly affected by the length of the documents [26].

In order to identify as many event-related documents as possible, a measurement to evaluate their relevance to the events of interest is necessary. The majority of existing research relies on the traditional TF-IDF text vector and distance measurements to assess the similarity [5, 10, 27]. Though TF-IDF, is widely used in Natural Language Processing as a measurement of words' importance and offers great performance for long paragraph text-mining, its accuracy for shorter tweets-alike document is still unsure [20]. In fact, Microblog posts are naturally unstructured with many colloquial

expressions and often do not comply with the normal syntax used in the Web. Only spare TF-IDF vectors can be formulated from tweets, which this is not a qualified input for traditional distance measurements. Recently, an attempt to associate tweet-level features with other metadata was conducted [28], however it still measured the event in a static way without considering the temporal evolution of the topics. In this paper, we propose to use a similarity measurement for a time series to overcome the above problem.

3 Social Microblogging Service: Twitter

Topic Indicator Conversational Hashtags. Twitter has sometimes been described as "the SMS of the Internet[7]" due to its conversational characteristic. This is supported by its well-known @ mention, RT retweet and # hashtag annotation. In this work, the hashtag annotation is of special interest as it allows users to indicate what the message is about when they publish a tweet [14]. By adding a # mark before the topic words, users can generate their own topic indicator at any moment. Twitter's user interface automatically associates a hyperlink for each hashtag to allow people to retrieve all tweets with the same hashtag in just a click. As the adaptive crawling framework is designed to collect data on a specific topic, this characteristic is adopted and explored.

Twitter API and Rate Limits. Twitter provides three public APIs to the developers and researchers for designing and implementation their desired tools: the Search API, the Streaming API and the Representational State Transfer (REST) API.[8] Of these, the Streaming APIs is used in our proposed model. This is because the Streaming API is the only interface that offers real-time access to the public tweets timeline. This API sends back 1 % of the whole tweets volume in its core database by using sample() function for each normal OAuth[9] enabled user. This 1 % limitation also applies to the filter method of the Streaming API. It is possible to use the method to generate a query to extract all tweets with specific criteria, e.g., keywords. However, the full amount of content is available only when the retrieved volume is less than 1 % of the total traffic of Twitter. Otherwise, that 1 % will be spread out across keywords, that is, only a subset of tweets will be retrieved for each individual keyword.

In the proposed adaptive crawling model, the filter method in the Streaming API is used to collect event relevant tweets. Twitter allows a maximum of 400 keywords for a single query and thus our search query was similarly limited.

[7] The SMS of the Internet: http://www.wisitech.com/blog/the-sms-of-the-internet.

[8] Twitter API Documentations: https://dev.twitter.com/.

[9] OAuth: http://oauth.net/.

4 Twitter Crawling Model Design

A Twitter crawler is a program that collects tweets or users' information through Twitter API matching a set of search criteria. In this section, a novel adaptive crawling model will be introduced. This adaptive crawling model is based upon the simple keyword crawler but embedded with a Keyword Adaptation Algorithm (KwA) running in real time.

4.1 Twitter Crawling Model

In this work, we are interested in keyword-based crawling, where every matching tweet will contain at least one of the defined search keywords. Compared with the simple (baseline) Twitter crawling model, the adaptive Twitter crawling model enables the adaptive crawling algorithm to leverage keyword adaptation in real-time.

4.1.1 Baseline Crawling

The baseline crawling model defines and uses a constant keywords set. In this model, a keywords set is used for focused crawling of a specific event. The keywords are manually defined according to the event of interest and remain unchanged for the entire collection period. The system flow of this crawling model is illustrated in Fig. 3 After sending the keywords with a query to the Twitter Streaming API, the qualified tweets will be returned as a stream. These tweets are stored in a database system. We use the dataset collected by this model as a ground truth in the evaluation section as this crawling model is used by most of the existing research.

4.1.2 Adaptive Crawling

The system structure of the adaptive crawling model is similar to the baseline crawling model for the Data Collection and Data Storage Components. The difference is the additional *Keyword Adaptation* component, as illustrated by Fig. 4. This component enables the application of the Simple Keyword Adaptation Algorithm (SKwA) and Refined Keyword Adaptation Algorithm (RKwA) described in the next section when crawling data in real-time events.

Fig. 3 System flow of simple Twitter crawling model (baseline)

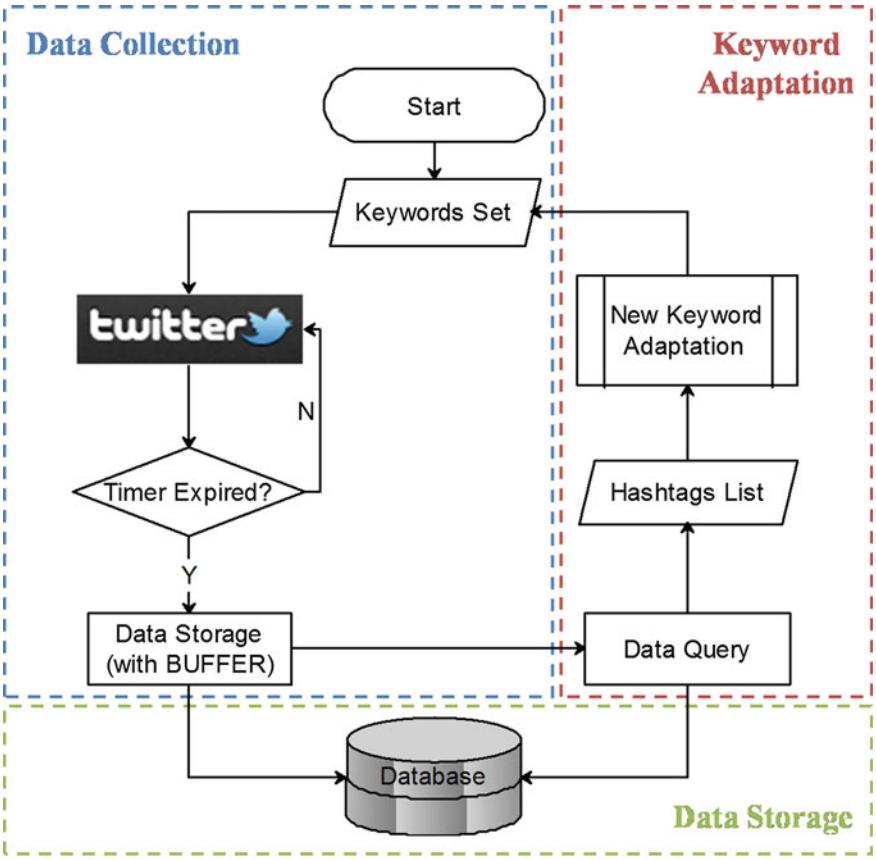

Fig. 4 System flow of the adaptive Twitter crawling model

In this model, the data collection process is triggered by the same set of predefined keywords as the baseline. The keyword adaptation feature enables the identification of popular event-related hashtags by using the Keyword Adaptation Algorithm (KwA). At the end of every time frame, the KwA is run over the previous time frame to generate a new keyword set. Finally, a query that encodes all the words in the keywords set is sent to the Twitter API and the time frame timer is restarted.

We exploit the traffic characteristics of hashtags gathered via Twitter Streaming API to realize keyword adaptation. Research shows that hashtags, a kind of user-defined index term that starts with #, have been used as topical markers to link relevant topics and events when people express their interests [14]. Exploiting hashtags for keyword searching not only reduces the complexity in getting the semantic meaning from tweets but also increases the efficiency of data analysis.

4.2 Keyword Adaptation Algorithm

The goal of a keyword adaptation algorithm is to automatically find the list of hashtags, beyond the initial set of keywords, appearing in tweets related to the event of interest. By using "automatic," we mean that keywords should be classified without manual intervention. Therefore, the essential problem is to figure out what kind of hashtags help with extra event-related information retrieval.

In our first attempt we apply the idea of trend detection in the Simple Keyword Adaptation algorithm. We assume the hashtags that appear frequently in tweets with initial keywords are related to the event. However, when evaluating Simple Keyword adaptation in the adaptive crawler we found that the Simple Keyword Adaptation algorithm introduces a lot of noise. Furthermore, due to the rate limit restriction from Twitter, the volume of the event-related tweets retrieved by this approach was far less than the volume collected by simply using the baseline crawler. This approach collected large amounts of noise.

In order to balance the efficiency and performance of crawling content under Twitter API restrictions, we designed the Refined Keyword Adaptation algorithm. In this section, complete details about both versions of Keyword Adaptation Algorithm (KwA) are presented.

4.2.1 Simple Keyword Adaptation Algorithm

In this SKwA, the collection of hashtags within the fixed time frame is represented as $H_{tf}(t_n) = \{h_1, h_2, \ldots\}$. The keywords set, sent to Twitter API in Fig. 3, at any time frame n, can be represented as $H(t_n) = \{h_1, h_2, \ldots\}$, where $h_k (k = 1, 2, \ldots)$ is an individual hashtag. Here, we use *keywords* to indicate hashtags that were eventually sent to Twitter for data collection. At the same time, the model also keeps two hashtags frequency lists, one for the whole collection period and the other for the current time frame. At the moment when any time frame n is passed, the hashtags frequency list for whole collection period is represented as $freq(t_n)$, while hashtags frequency list for the nth time frame is written as $freq_{tf}(t_n)$. The frequency list for the whole collection period updates every time frame, while the other list updates within the time frame when a new tweet arrives. The hashtag list and the frequency list have a one-to-one correspondence, i.e., the frequency count of a hashtag h_k at nth time frame is $freq_{tf}^{h_k}(t_n)$. The Frequency List Update algorithm is defined in Algorithm 1.

Apart from these two frequency lists, a minimum frequency ($freq_{\min}$), as a threshold for being a keyword, and an array of blacklist hashtags (H_{black}) are also used in the simple adaptive crawler to help with adaptation. The pseudocode in Algorithm 2 details this version of the Keyword Adaptation Algorithm (KwA).

This algorithm keeps at most $N = 400$ keywords for querying Twitter every 10 min, where N is the maximum number of hashtags in keywords set. When a new hashtag appears, the algorithm will check whether or not it already exists in the keywords set $H(t_n)$. If it is a query keyword, its whole period frequency list is incremented by 1. Otherwise, the hashtag is stored in the time frame hashtags list

Algorithm 1 Frequency List Update

Require: $H_{tf}, freq_{tf}^{h_k}$
1: **for** $\forall h$ in the incoming tweets **do**
2: **if** $\exists h_k = h : h_k \in H_{tf}(t_n)$ **then**
3: $freq_{tf}^{h_k}(t_n) = freq_{tf}^{h_k}(t_n) + 1$;
4: **else**
5: $H_{tf}(t_n) = H_{tf}(t_n) + 1$;
6: $freq_{tf}^{h_k}(t_n) = 1$
7: **end if**
8: **end for**

Algorithm 2 Simple Keyword Adaptation (SKwA)

Require: $H_{tf}, freq_{tf}^{h_k}$
1: **for** $\forall h \in H_{tf}(t_n)$ **do**
2: **if** $h \in H_{blacklist}$ **or** $freq_{tf}^{h_k}(t_n) < freq_{min}$ **then**
3: $H_{tf}(t_n) = \{h_k | h \in H_{blacklist}, h_k \neq h\}$;
4: $freq_{tf}(t_n) = \{freq_{tf}^{h_k}(t_n) | freq_{tf}^{h_k}(t_n) \in freq_{tf}(t_n), h_k \neq h\}$
5: **else**
6: $H(t_n) = H(t_{n-1}) \cup \{h_k | freq_{tf}^{h_k}(t_n) \in Top\ n(freq_{tf}^{h_k}(t_n))\}$;
7: $freq(t_n) = freq(t_{n-1}) \cup \{freq_{tf}^{h_k}(t_n) \in Top\ n(freq_{tf}^{h_k}(t_n))\}, wheren = N - num[H(t_{n-1})]$
8: **end if**
9: **end for**

temporarily. When the timer expires, hashtags in the time frame hashtags list are sorted according to their frequency. Top ones will be added to the keywords set. In other words, hashtags with a low frequency within time frame n do not become a keyword.

This SKwA employs three noise reduction steps to avoid overwhelming the new keyword set with non-related keywords. First, the threshold for being a keyword, $freq_{\min}$, helps to filter out the unusual hashtags. While those hashtags can be relevant to the event of interest, they are not worthy of collection because they only generate a tiny amount of traffic. In addition, the introduction of these low frequency hashtags will significantly increase the calculation cost, both in space and time. As a result, we set the $freq_{\min}$ with an empirical value to be once per minute. Second, by discarding the long term, low frequency items, the crawler can improve the utility of n keywords. This mechanism functions as follows: for hashtag h has low values for a long period ($freq_{tf}(t_n), freq_{tf}(t_{n-1}), freq_{tf}(t_{n-m})$), it will be removed from the keywords set. Lastly, the introduction of a keyword blacklist allows noisy keyword to be manually filtered. The blacklist is empty when the crawler is started. Users can identify and add non-related words to the blacklist during the collection period. The algorithm will check this list every time it identifies new search terms so that it can discard the words that are in the blacklist. For the experiments in this paper, the blacklist words belong to either general association with news channels (e.g., BBC and CNN) or as hashtags used by follow up and follow back activities (e.g., teamfollow and followback).

4.2.2 Refined Keyword Adaptation Algorithm

Our initial attempts show that extra traffic can be produced when using the proposed SKwA when running with the adaptive crawler. However, we found the dataset collected through SKwA also contains a large amount of non-related tweets: the longer the crawler runs, the larger the proportion of noisy tweets. The noise, i.e., non-related tweets, eventually overwhelm the event-related data, which results in a chaotic dataset. This issue is caused by the fact that the algorithm relies on the collected content: a clean dataset will help the crawler to better adapt; a noisy dataset always becomes even noisier.

In order to reduce the impact of noisy information on the adaptive dataset, the traffic pattern of hashtags is exploited to classify those potential keywords according to their relevance to the events. The problem is how to modify the SKwA so that the adaptive crawler collects a greater amount of highly event associated data without significantly increasing the dataset noise.

The refined version first automatically gets a hashtags list based on the SKwA. The list is then passed to an extended part of the keyword adaptation algorithm for assessing the elements' relevance to the event. Here, we introduce the *correlation coefficient* to evaluate the relevance. In order to calculate the correlation between two hashtags, we subdivide the time frame into several time slots. The frequency counts of each time slot is represented by $freq_{tf}^{h_k}(t_n)$. This array indicates the frequency counts of a hashtag h_k in all the time slots within the nth time frame. The collection of initial keywords is represented as $H_{seed} = \{h_1, h_2, \ldots\}$. Instead of using $H(t_n)$, the keywords set will be sent to Twitter API at the end of each time frame and is written in the form $H_{fin}(t_n)$, $H(t_n)$. It is a temporal list which holds the same result as that used by SKwA. The pseudocode is updated as Algorithm 3.

Algorithm 3 Refined Keyword Adaptation (RKwA)

Require: $H_{seed} = H(t_n) \cup H_{fin}(t_{n-1})$, $H_{fin}(t_n) = H_{BL}$
1: **Execute Algorithm 2 SKwA**
2: **for** $\forall h_x \in H_{seed}$ **do**
3: **for** $\forall h_y \in H_{fin}(t_n)$ **do**
4: **if** $h_y \in H_{BL}$ **and** $cor(freq_{tf}^{h_x}(t_n), freq_{tf}^{h_y}(t_n)) > Thres_1$ **then**
5: $H_{fin}(t_n) = \{h | h \in H_{fin}(t_n)$ **or** $h = h_x\}$;
6: **else if** $h_y \neq H_{BL}$ **and** $cor(freq_{tf}^{h_x}(t_n), freq_{tf}^{h_y}(t_n)) > Thres_2$ **then**
7: $H_{fin}(t_n) = \{h | h \in H_{fin}(t_n)$ **or** $h = h_x\}$;
8: **end if**
9: **end for**
10: **end for**

The initial keys H_{seed} and correlation measurements *cor* are defined based on the following hypotheses:

Hypothesis 1 (*H1*): *the initial keywords used for both baseline crawler and adaptive crawler are the most representative words that describe the event of interest.*

Hypothesis 2 (*H2*): *trending keywords for an event during one particular or several sequential time frames are likely to exhibit similar traffic patterns.*
Hypothesis 2.1 (*H2.1*): *the frequency of occurrence of two trending keywords shows a linear relationship. Namely, when keyword A appears more, the frequency of keyword B will also increase, and vice versa.*

Consequently, the initial keywords used by the baseline crawler and adaptive crawler with SKwA are also selected as initial keys in RKwA. A popular linear correlation measurement, i.e., Pearson correlation, which is defined by the following equation, is chosen as the measurement of similarity between related keywords.

$$cor = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{1}$$

Here, sequence X represents the $freq_{tf}^{h_x}(t_n)$ and Y for the $freq_{tf}^{h_y}(t_n)$ in the algorithm. That is to say, hashtag $h \in H_{tf}(t_n)$, as calculated by SKwA, is only retained in RKwA if it has a high correlation with one of the seed keywords. For example, #100aday is a trending hashtag but irrelevant to the event. It was detected as a keyword by SKwA, but it was successfully excluded by RKwA because of its low correlation to the initial hashtag.

The threshold values for $Thres_1$ and $Thres_2$ also need to be set for executing RKwA. We use a single variable approach to choose their values: one of the thresholds was fixed while the other was changing gradually. We found that changing of $Thres_1$ did not bring too much impact on the result, but the differences introduced by changing of $Thres_2$ is notable as there is always a range of threshold values that can make the signal-to-noise ratio higher than others. Therefore, the final value we choose is $Thres_1 = 0.5$ and $Thres_2 = 0.8$.

5 Evaluation of Adaptive Crawling Model

The purpose of the evaluation is to test if the proposed adaptive crawling model helps to collect additional data without introducing too much noise.

In our previous work [29], the experiments were conducted on a historical dataset of the 2012 London Olympic Games and with only a theoretical analysis; whereas here we apply both SKwA and RKwA to our adaptive crawling model and test them with a large public event in real-time. Our aim is to demonstrate that the information gain is at the same level as we showed, and the signal-to-noise ratio (i.e., ratio between the event related information and event irrelevant information) is much more significant than our previous estimation.

In more detail, an overview of the collected datasets, including one for baseline crawler and one for adaptive crawler with SKwA and RKwA, respectively, is described first. Then, we analyze and evaluate the proposed adaptive crawling

Table 1 Tweet volume generated by different crawling approaches

	Baseline	SKwA	RKwA
Tweet count	550,417	10,433,355	2,472,953
Unique tweet	10,275 (1.8 %)	9,534,735 (91.4 %)	1,252,577 (50.6 %)

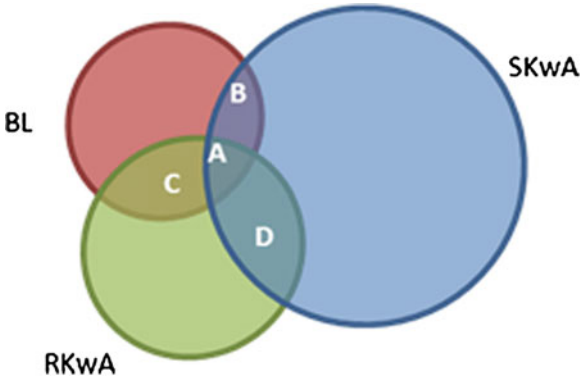

Fig. 5 Number of common tweets in baseline, SKwA and RKwA datasets: $A = 21{,}4218$, $B = 2{,}084$, $C = 323{,}840$, $D = 682{,}318$

model by classifying the retrieval keywords and tweets. Accordingly, the relevance of keywords and that of tweets from all the three datasets will be assessed with a quantitative method.

5.1 Dataset Overview

The datasets were collected during the 2013 Glastonbury festival[10] period. Three crawlers were run for the tweets collection, first the baseline crawler, and then the two instances of the adaptive crawler, with the SKwA and RKwA respectively. Only "Glastonbury" is used as the initial keyword for all the three crawlers.

Table 1 and Fig. 5 illustrate the tweet volume collected for "Glastonbury" from 2013-6-28, 19:00:00, BST to 2013-7-1, 07:00:00, BST. The collection period lasted 60 hours, with more than half a million tweets collected from the baseline crawler alone. The number of tweets collected by SKwA is almost 20 times the number collected by the baseline crawler. In Table 1, the column "unique" is the number of tweets that appear only in that dataset. Provided that all the crawlers start with the same initial keyword "Glastonbury", SKwA and RKwA datasets should contain all the tweets in the baseline dataset, i.e., the cell indicating the unique number of tweets in the baseline dataset should be zero. However, some of the tweets, even if they contain the initial keywords, cannot be retrieved by the SKwA due to the 1 % rate limitation. When the number of keywords increases, the volume of tweets containing those keywords also increases and is more likely to exceed the rate limit.

[10] What is Glastonbury: http://www.glastonburyfestivals.co.uk/information/what-is-glastonbury.

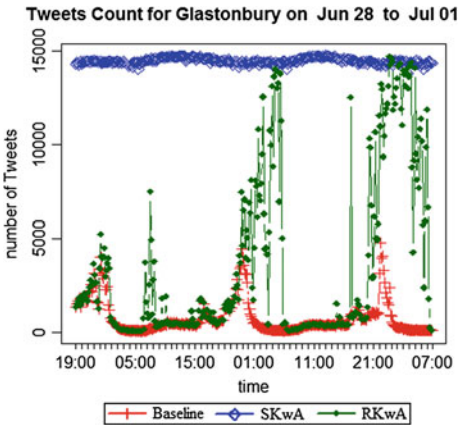

Fig. 6 Tweet volume for Glastonbury festival (10 min interval)

Figure 6 shows the traffic volume, every 5 min for each of the three datasets, and provides a graphical view for examining the period when any of the crawlers hit the Twitter rate limits. According to the figure, it is obvious that the number of tweets approached 2,900/min in the SKwA dataset. Based on an empirical test, this value is close to the upper tweet volume limit when accessing the Streaming API free of charge. Due to the reason that SKwA crawler is always rate limited, tweets showing up in the baseline dataset can be lost in the SKwA dataset, and therefore result in the unique tweets in baseline dataset. Compared with the SKwA dataset, the RKwA dataset contains many more tweets than the baseline dataset, i.e., almost all the tweets in baseline dataset also showed in RKwA dataset. It also included some tweets from SKwA and 50 % unique tweets. Though the RKwA crawling also hit the rate limit at some points according to Fig. 6, it still achieved an acceptable performance most of the time.

5.2 Evaluating the Keyword Adaptation Algorithm

The previous section qualitatively illustrates the overall statistics of the three datasets. The next step is to quantitatively analyze to show that the adaptive crawling model helps to extract extra relevant event information. This section details the evaluation procedures for revealing whether or not the extra tweets are all related to the event in question.

5.2.1 Evaluation Setup

The aim of this experiment is to verify that RKwA performs better than SKwA in retrieving a greater amount of event-related information while retaining the noise

(non-related tweets)-to-signal (event-related tweets) ratio at a low level. The following hypothesis acts as a condition for evaluating the performance of SKwA and RKwA:

Hypothesis 3 (*H3*): *a tweet is likely to only be about one topic which is described by hashtags, and therefore its correlation to an event of interest is determined by its hashtags.*

H3 determines whether or not the tweet's hashtags affect the tweet's relevance to an event of interest. Based on this hypothesis, we design the procedures for evaluating the performance of the adaptive crawling model as follows:

5.2.2 Labeling Keywords Manually

In order to filter out noisy tweets, the first step is to distinguish between the related and non-related keywords by manually labeling: hashtags shown in the keywords set are manually classified into corresponding categories. Three independent participants are involved in this labeling process. The final result is based on the average produced by two independent participants. A third labeller was introduced in the case of a disagreement.

Hashtags in different time periods were labelled according to how closely they are related to the Glastonbury Festival. For example, "#glasto2013" is definitely related, while "#6hobbs" is more complicated to classify. It could be related since it represents a program for BBC Radio Music which always broadcasts information about music. However, it may also include information other than the Glastonbury Festival. In our grading strategy, it was classified as possibly-related. All the hashtags were labelled into five categories based on the criteria in Table 2.

Table 2 The hashtag category and grading tratego

Hashtag category	Specification	Score
Related (C1)	Hashtags contain the terms Glastonbury, band names or song names that appear during the festival	+2
Possibly-related (C2)	Hashtags stand for media which broadcasts the event, as well as emotional hashtags and those emerged with ongoing affairs (nextyear, best-seats, etc.)	+1
Non-related (C3)	Hashtags showing no particular relationship with the event	−2
Not known (C4)	Non-English hashtags that the manual taggers did not identify	0
Non-keyword hashtags	Hashtags that have not been selected as keywords	−1

5.2.3 Classifying Tweets According to the Manual Labeling

In this step we classify whether or not a tweet is related to the event based on the hashtags it contains using the grading system in Table 2. Each hashtag is assigned a score and the final grade of a tweet is the sum of all the hashtags' scores.

By using this strategy, tweets with a grade more than 0 are classified as related tweets, and those less than or equal to 0, as non-related tweets. The grading system can identify non-related tweets even if it carries related hashtags. For example, "*Friday night! Meet new people—FREE! onclique #meetingpeople #Bristol #instagram #Glastonbury #Manchester #onclique*" are classified as non-related tweet. The final grade is −2 because the score introduced by #Glastonbury is cancelled out by other negative instances.

Therefore, the baseline, SKwA and RKwA datasets were all classified into two sub-datasets, related and non-related tweets datasets. Finally, we compare the proportion of related and non-related tweets in all datasets to check the levels of noise introduced and the proportion of event-related information retained.

5.3 Experiment Results

A subset of the Glastonbury data was selected for the evaluation. The test set is during 8:00 29 June to 00:00 30 June as this is the period where both the SKwA and RKwA worked properly with normal behavior (e.g., not suddenly polluted by noise).

The tweets volume of all the crawlers during the selected period is shown in Fig. 7. According to this figure, the first fluctuation appeared at 16:00, while the highest traffic period started at night from about 20:00, and reached peak at about 23:00. This is because the famous music performers started to show up in the afternoon and

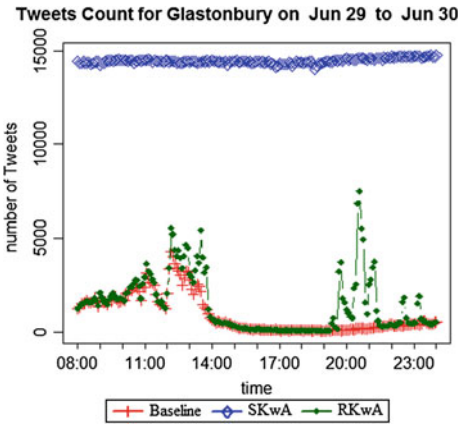

Fig. 7 Tweet volume (Evaluation period) for Glastonbury festival

the performances finished at midnight. It is clear that the adaptive crawler with the SKwA was always rated limited, while the other adaptive crawler identified extra tweets during the peak period, when compared with the baseline crawler.

5.3.1 Relevance of Identified Topical Keywords

In Table 3, each row is the number of keywords in the corresponding category. The first column describes the keyword composition for the baseline crawler (BL). The value 1 shows in the first category and indicates that it only maintains single keywords during the whole crawling period. Namely, the baseline crawler does not adapt the keyword set. According to the figures here, the SKwA did provide an extra 30 ($15 + 16 - 1$) event keywords. But clearly, its retrieval keyword set is very noisy as C3 keywords dominate most of the SKwA keywords set. The statistics in the third column shows that the RKwA performs much better than the SKwA. The last column makes this clear: it is the RKwA-to-SKwA (RS) ratio between the number of $C_x (x = 1, 2, 3, 4)$ keywords from the RKwA crawler to that from the SKwA crawler. It is clear that the RKwA reduced the proportion of C3 keywords in the SKwA dataset by more than a thousand, i.e., the noisy keywords are dropped to only 7.06 %. Meanwhile, the RKwA identified more C1 and C2 keywords compared with the SKwA. The RS ratio for C1 and C2 keywords reached 440 and 137.5 % respectively. This indicates that by using the proposed RKwA, the event-related terms are more likely to be identified, while the introduction of noisy keywords is controlled. This provides preliminary evidence that the RKwA performs much better than the SKwA.

In addition, the extra event-related keywords can pull extra event content, especially for the RKwA dataset, as shown in Fig. 8a. The volume of the band name keyword "*rollingstones*" has its peak at the same time for all the three crawlers, though the volume varies. The difference in information gain between the baseline and the RKwA crawling illustrates that the adaptive crawling has the potential to fetch additional event-related information. More specifically, the RKwA dataset contains more tweets with *#rollingstones* than either the SKwA dataset or the baseline dataset. Surprisingly, the SKwA dataset maintains the lowest volume of tweets containing "*rollingstones*". Considering that the SKwA adaptive crawler was rate limited all the time and collected tweets with more keywords, this phenomenon is caused by the spread of the space of other non-related traffic. Apart from this, the

Table 3 Hashtag count of manually labeling categories

Keywords count category	BL	SKwA	RKwA	RKwA to SKwA ratio (%)
Related (C1)	1	15	66	440.00
Possibly-related (C2)	0	16	22	137.50
Non-related (C3)	0	1,360	96	7.06
Not known (C4)	0	500	30	6.00
Total	1	1,891	214	11.32

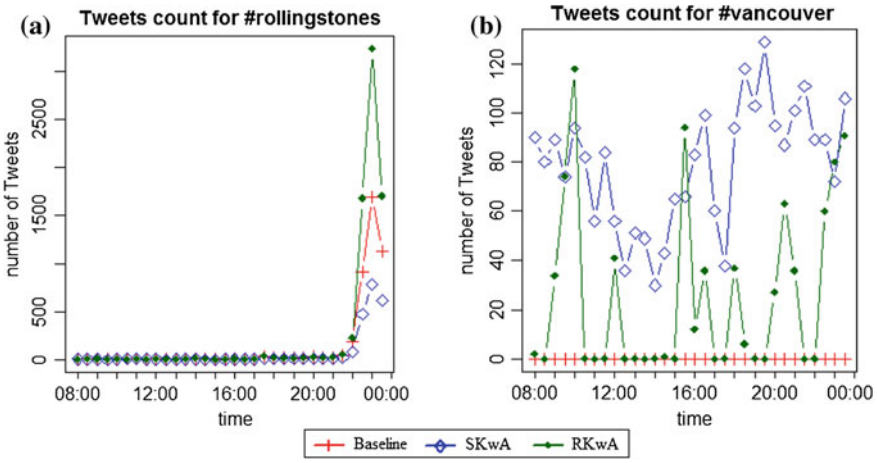

Fig. 8 Comparison of event-related versus event-irrelevant keywords in the three datasets. **a** Trending event keyword *#rollingstones*. **b** Event irrelevant keyword *#vancouver*

apparent differences for the volume of tweets containing "*#vancouver*" in the SKwA dataset and the RKwA dataset in Fig. 8b proved that the RKwA is also able to reduce the impact of irrelevant keywords. Random spikes of "*#vancouver*" for the RKwA dataset shown in Fig. 8b are introduced by tweets that carry both the event-related keywords and the "*vancouver*". However, the event-related keyword in such tweets is not "*glastonbury*", so it cannot be retrieved by the baseline crawler. For example, when the RKwA adaptive crawler retrieved tweets for the keyword "*bbcglasto*", tweet like "Trending *#rollingstones #bbcglasto #vancouver #chic* #followme" was also collected.

5.3.2 Relevance of Collected Tweets

The original intention of proposing this adaptive crawler is to fetch extra event-related tweets. We examine this over all datasets. Specifically, tweets from the SKwA and RKwA datasets were classified according to the final grade. In the baseline tweets classification task, the final grade is calculated by referring to the manually labelled results of the other two adaptive datasets. If the baseline tweets contain any of the labelled keywords, the same value will be added to or deducted from the final grade.

Figure 9 shows the traffic volume of irrelevant event tweets and the event-related tweets in these three different datasets. Figure 9a illustrates that SKwA introduces a great amount of noise—most of the traffic (about 94 %) from SKwA dataset is irrelevant to the event. The green line for RKwA clearly illustrates that the RKwA crawler performs well in reducing the amount of noise. Compared with the SKwA dataset, the irrelevant event tweets in the RKwA dataset are relatively few. According to Fig. 9b, the SKwA only introduced extra event content at the beginning of the evaluation period. At the time when the event content is increasing, i.e., after

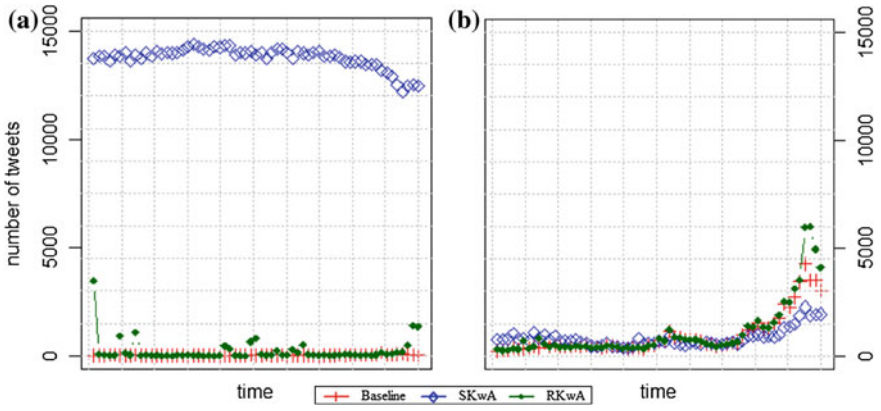

Fig. 9 Proportion of tweets and their relevances to the event in the three datasets **a** Irrelevant event tweets. **b** Event-related tweets

20:00, the SKwA loses a large proportion of event content even compared with the baseline one. Because of the rated limited condition, the amount of tweets that can be fetched is fixed. When most of the collection channel was occupied by the event-irrelevant tweets, the amount of event-related tweets is significantly reduced. On the other hand, the RKwA performs well on collecting extra event-related information: compared with baseline crawling, the RKwA crawler fetched more than 70 % extra event-related information at the event peak. On average, the RKwA can identify about 100 more event tweets every minute compared to baseline crawling. These extra event-related tweets give additional event information. The observation here is that RKwA performs well. It supplies extra event-related tweets while reducing the noise in the SKwA.

One interesting phenomenon is that there is also a small amount of noise in the baseline dataset. Although the word "*glastonbury*" is highly specific to the festival, it also introduces noise because there are tweets that contained Glastonbury but (1) did not talk about the event itself; (2) were spam tweets. The proposed grading strategy is not always successful in tackling the first problem but is designed to deal with the second one. It works well for spam tweets that are published to spread trending topic and hashtags. These kinds of tweets contain many hashtags without any plain text or content. The grading strategy can identify them by reducing the final score when non-keyword hashtags appear. This kind of spam is one of the most prevalent sources of the irrelevant tweets in the baseline dataset.

6 Conclusion

In this paper, we focus on finding a solution for crawling Microblog feeds in real-time. By exploiting the hashtags from Twitter feeds, we proposed an adaptive crawling model that reviews the retrieved content to identify new keywords for automatic live

event tweets collection. In order to improve the reliability and robustness, we further refined the KwA to support higher precision. Based on the evaluation results, we have shown that:

- The trend detection-based SKwA is not efficient enough to identify event keywords for adaptive crawling, as it introduces too much noise;
- The RKwA performs well in reducing non-related keywords, and distinguishes an extra amount of the event-related keywords from the noisy hashtags;
- The adaptive crawler based on RKwA is able to collect extra event-related tweets (70 %) compared to the baseline crawling approach, while maintaining a noise level below 35 tweets per minute.

The future work for this adaptive crawling model includes an improvement of the new keyword selection schema and the use of an auto initial seed setup. Currently, the threshold value for the correlation is set to be a fixed value. If the system can automatically update the thresholds without losing the real-time efficiency, the performance will be more stable. Also, this can reduce the chance of hitting the rate limit. Another improvement regarding the keyword selection is the automatic selection of baseline keywords, i.e., initial seeds. Furthermore, research toward identifying and validating additional metrics for accessing the adaptation is also a goal of our future research. The aim is to combine other additional metrics with the RKwA to improve the performance of our adaptive crawler.

References

1. Zhao D, Rosson MB (2009) How and why people Twitter: the role that micro-blogging plays in informal communication at work. In: Proceedings of the ACM 2009 international conference on supporting group work (GROUP'09), pp 243–252
2. Sakaki T, Okazaki M, Matsuo Y (2010) Earthquake shakes Twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on world wide web (WWW'10), pp 851–860
3. Starbird K, Palen L (2012) (How) will the revolution be retweeted?: information diffusion and the 2011 Egyptian uprising. In: Proceedings of the ACM 2012 conference on computer supported cooperative work (CSCW'12), pp 7–16
4. Becker H, Iter D, Naaman M, Gravano L (2012) Identifying content for planned events across social media sites. In: Proceedings of the fifth ACM international conference on web search and data mining (WSDM'12), pp 533–542
5. Chakrabarti D, Punera K (2011) Event summarization using tweets. In: Proceedings of the 5th international AAAI conference on weblogs and social media (ICWSM'11), pp 66–73
6. Liu SB, Palen L (2009) Spatiotemporal mashups: a survey of current tools to inform next generation crisis support. In: Proceedings of the 6th international conference on information systems for crisis response and management (ISCRAM'09)
7. Krishnamurthy B, Gill P, Arlitt M (2008) A few chirps about twitter. In: Proceedings of the first workshop on online social networks (WOSN'08), pp 19–24
8. Tumasjan A, Sprenger TO, Sandner PG, Welpe IM (2010) Predicting elections with twitter: what 140 characters reveal about political sentiment. In: Proceedings 4th international AAAI conference on weblogs and social media (ICWSM'10), pp 178–185

9. Abel F, Celik I, Houben G-J, Siehndel P (2011) Leveraging the semantics of tweets for adaptive faceted search on twitter. In: Proceedings of the 10th international conference on the semantic web (ISWC'11), pp 1–17
10. Bifet A, Holmes G, Pfahringer B (2011) MOA-TweetReader: real-time analysis in Twitter streaming data. In: Proceedings of the 14th international conference on discovery science (DS'11), pp 46–60
11. Petrovi S, Osborne M, Lavrenko V (2010) Streaming first story detection with application to Twitter. In: Human language technologies: the 2010 annual conference of the North American chapter of the association for computational linguistics (HLT'10), pp 181–189
12. Huberman BA, Romero DM, Wu F (2008) Social networks that matter: Twitter under the microscope
13. Liang F, Qiang R, Yang J (2012) Exploiting real-time information retrieval in the microblogosphere. In: Proceedings of the 12th ACM/IEEE-CS joint conference on digital libraries (JCDL'12), pp 267–276
14. Tsur O, Rappoport A (2012) What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities. In: Proceedings of the fifth ACM international conference on web search and data mining (WSDM'12), pp 643–652
15. Kwak H, Lee C, Park H, Moon S (2010) What is Twitter, a social network or a news media. In: Proceedings of the 19th international conference on world wide web (WWW'10), pp 591–600
16. Naghavi M, Sharifi M (2012) A proposed architecture for continuous web monitoring through online crawling of blogs. Int J UbiComp 3(1):11–20
17. Lanagan J, Smeaton A (2011) Using Twitter to detect and tag important events in sports media. In: Proceedings of the fifth ACM international conference international AAAI conference on weblogs and social media (ICWSM'11)
18. Nichols J, Mahmud J, Drews C (2012) Summarizing sporting events using Twitter. In: Proceedings of the 2012 ACM international conference on intelligent user interfaces (IUI'12), pp 189–198
19. Yin J, Lampert A, Cameron M, Robinson B, Power R (2012) Using social media to enhance emergency situation awareness. IEEE Intell Syst. 27(6):52–59
20. Perez-Tellez F, Pinto D, Cardiff J, Rosso P (2010) On the difficulty of clustering company tweets. In: Proceedings of the 2nd international workshop on search and mining user generated contents (SMUC'10), pp 95–102
21. Kontostathis A, Galitsky L, Pottenger WM, Roy S, Phelps DJ (2003) A survey of emerging trend detection in textual data mining. In: Michael W Berry (ed) Survey of Text mining. Springer-Verlag, New York, pp 185–224
22. Mathioudakis M, Koudas N (2010) TwitterMonitor: trend detection over the twitter stream. In: Proceedings of the 2010 ACM SIGMOD international conference on management of data, Indianapolis, 06–10 June 2010
23. Cataldi M, Caro LD, Schifanella C (2010) Emerging topic detection on Twitter based on temporal and social terms evaluation. In: Proceedings of the 10th international workshop on multimedia data mining, Washington, 25–25 July 2010, pp 1–10
24. AlSumait L, Barbar D, Domeniconi C (2008) On-line LDA: adaptive topic models for mining text streams with applications to topic detection and tracking. In: Proceedings of the 2008 eighth IEEE international conference on data mining (ICDM'08), 15–19 Dec 2008, pp 3–12
25. Lau JH, Collier N, Baldwin T (2012) On-line trend analysis with topic models: #twittertrends detection topic model online. In: Proceedings of the 24th international conference of on computational linguistics, pp 1519–1534
26. Hong L, Davison BD (2010) Empirical study of topic modeling in Twitter. In: Proceedings of the first workshop on social media analytics (SOMA'10), ACM, New York, pp 80–88
27. Varga A, Cano AE, Ciravegna F (2012) Exploring the similarity between social knowledge sources and twitter for cross-domain topic classification. In: Proceedings 11th international semantic web conference on knowledge extraction and consolidation from social media (ISWC 2012)

28. Abhik D, Toshniwal D (2013) Sub-event detection of natural hazards using features of social media data. In: International world wide web workshop on social web for disaster management (SWDM'13), Rio de Janeiro, Brazil
29. Wang X, Tokarchuk L, Cuadrado F, Poslad S (2013) Exploiting hashtags for adaptive microblog crawling. In: IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM 2013)
30. Baray MB, Kurt H, On-line new event detection and tracking in a multi-resource environment. Unpublished master's thesis, Bilkent University, Computer Engineering Department
31. Byun C, Kim Y, Lee H, Ko Kim K (2012) Automated Twitter data collecting tool and case study with rule-based analysis. In: Proceedings of the 14th international conference on information integration and web-based applications and services (IIWAS'12), pp 196–204
32. Boanjak M, Oliveira E, Martins J, Rodrigues EM, Sarmento L (2012) TwitterEcho: a distributed focused crawler to support open research with Twitter data. In: Proceedings of the 21st international conference companion on world wide web (WWW'12 Companion), pp 1233–1240

Comparison of Emoticon Recommendation Methods to Improve Computer-Mediated Communication

Yuki Urabe, Rafal Rzepka and Kenji Araki

Abstract This paper describes the development of an emoticon recommendation system based on users' emotional statements. In order to develop this system, an innovative emoticon database consisting of a table of emoticons with points expressed from each of 10 distinctive emotions was created. An evaluation experiment showed that our proposed system achieved an improvement of 28.1 points over a baseline system, which recommends emoticons based on users' past emoticon selection. We also integrated the proposed and baseline systems, leading to a performance improvement of approximately 73.0 % in the same experiment. Evaluation of respondents' perceptions of the three systems utilizing an SD scale and factor analysis is also described in this paper.

Keywords Emoticon · Affect analysis · Recommendation method · Smartphone application

Y. Urabe (✉) · R. Rzepka · K. Araki
Hokkaido University, Graduate School of Information Science and Technology, Sapporo, Japan
e-mail: y_urabe@media.eng.hokudai.ac.jp

R. Rzepka
e-mail: rzepka@ist.hokudai.ac.jp

K. Araki
e-mail: araki@ist.hokudai.ac.jp

Ö. Ulusoy et al. (eds.), *Recommendation and Search in Social Networks*, Lecture Notes in Social Networks, DOI 10.1007/978-3-319-14379-8_2

1 Introduction

Social Network Services (SNS) have grown rapidly throughout the world, such as Facebook[1] and Twitter,[2] which now handle 1.19 billion[3] and 232 million[4] monthly active users, respectively. Such services have dramatically increased social user interaction on the Internet in comparison to the days when only email and online chatting systems existed. However, due to a lack of nonverbal cues such as facial expressions, body movements, and emotional tones, computer-mediated communication (CMC) often fails to present personal dispositions that are transparently expressed in face-to-face communication. These nonverbal cues account for about 93 % of our daily communication [1], a fact that we should not ignore. Hence, users compensate for these shortcomings by using emoticons.

Emoticons are composed of letters and symbols and represent facial marks or movements. These emoticons can be divided into two styles: a horizontal style (e.g., "(^ _ ^)") and a vertical style (e.g., ":)"). The horizontal style is especially popular in Asian countries such as Japan, South Korea, and China, while the vertical style is mainly used in Western countries [2]. The number of emoticons in the horizontal style is increasing day by day, so much that a Japanese online emoticon dictionary [5] now includes more than 58,000 different types of emoticons, while the vertical type only consists of around 260 emoticons. These emoticons are sophisticated enough to express users' feelings and intentions in CMC; therefore, they are added to sentences in order to express intentions that cannot be expressed by words alone, to enhance the sentence and to express sarcasm and humor [3, 4]. Users insert emoticons by creating them on their own using keypads and keyboards, copying and pasting from online emoticon dictionaries or from emoticon dictionaries installed in devices like smartphones. However, these approaches are not efficient, because many symbols and letters are not simple to type. For example, 58,000 emoticons described in the previous paragraph contain only about 23.6 % of letters and symbols that can be entered from a computer keyboard. Also, choosing one emoticon from emoticon dictionaries that contain hundreds or thousands of emoticons is extremely inconvenient. In order to solve these problems, we propose an emoticon recommendation method that recommends emoticons according to an emotion type analyzed from users' statements. As Kato et al. [5] demonstrated in his research that emoticons are chosen depending on the valence of input (i.e., positive emoticons are chosen with positive contexts, and vice versa), we believe that recommending emoticons depending on the emotion type of the input would be very useful to users.

---

[1] https://www.facebook.com/.

[2] https://twitter.com/.

[3] http://thenextweb.com/facebook/2013/10/30/facebook-passes-1-19-billion-monthly-active-users-874-million-mobile-users-728-million-daily-users/, retrieved on Nov. 25, 2013.

[4] http://www.businessinsider.com/one-half-of-twitters-active-users-tweet-monthly-2013-11, retrieved on Nov. 25, 2013.

[5] http://www.kaomoji.sakura.ne.jp/, retrieved on Nov. 25, 2013.

Our proposed system utilizes two main features: an affect analysis system, ML-Ask [6], and an originally created emoticon database. Our emoticon database contains 59 emoticons, each emoticon showing the extent of each of 10 distinctive emotions (joy/delight, anger, excitement, sadness/gloom, fear, fondness/liking, relief, shyness, surprise/amazement, dislike) on a 5-point scale. We performed a comparison experiment of our proposed method and a baseline method used in the Japanese keypad in iOS. The baseline method recommends emoticons according to the user's past selections. An experiment proved that participants chose emoticons that were among the top five of those recommended by our proposed system, at 28.1 points higher than that of the baseline system. Also, the result was improved to approximately 73.0 % (an improvement of 43.5 points over the baseline method) in the same experiment when we integrated both methods. We also discovered that users' attitudes toward the integrated system and the proposed system were more positive than the baseline system, by conducting evaluation using the semantic differential (SD) method and factor analysis.

2 Related Works

In the field of sentiment analysis, Ptaszynski et al. [7] created an affect analysis system for emoticons: "CAO". "CAO" extracts an emoticon from a sentence and analyzes the specific emotion type according to the theory of kinesics. This system is capable of analyzing more than three million emoticons. Additionally, Emura and Seki [8] proposed an emoticon recommendation method based on the estimation of emotions, communication types, and action types written by users. This research revealed the importance of recommending emoticons according to not only the emotion type provided by the input but also communication types (e.g., greetings and gratitude), and action (e.g., sleep, run, etc.), achieving 66.7 % suitable emoticon recommendations to users. The emoticons in the databases of these systems were gathered from online dictionaries, wherein emoticons are categorized to certain emotion types by administrators, but it has not yet been assessed whether these express the correct emotion types. Meanwhile, Kawakami [9] created an emoticon database which is numerically categorized according to certain emotion types. Kawakami concentrated on how much an emoticon expresses each emotion and investigated how much the emotion emphasizes the sentence. The research revealed that some emoticons express plural emotion types strongly.

In order to create an emoticon database for our proposed system, we employed Kawakami's [9] work in order to develop a more accurate emoticon recommendation system. Creating a database of emoticons showing a numerical expression of each emotion could be a step toward the creation of a system that can recommend emoticons that express the users' complicated emotional state.

3 Emoticon Recommendation Method

The system utilizes two main procedures (Fig. 1). First, the system analyzes the emotion in the user input. We used an affect analysis system, ML-Ask [6] (More details of the ML-Ask are described on 3.1). Second, the system rearranges the emoticon database in the order of suitability to the emotion specified by ML-Ask and recommends the emoticons from top of the list to the user. We created the emoticon database originally by performing a survey of 60 Japanese university students. Next, the user chooses an emoticon that matches the input (the system accordingly registers the frequency of the chosen emoticon in the database, incrementing by one each time an emoticon is selected). Lastly, the system inserts the emoticon right after the input. We implemented the procedure on the iPhone (iOS 7.0) (Fig. 2).

3.1 ML-Ask

Ptaszynski et al. [6] developed ML-Ask for analyzing emotions from Japanese texts. ML-Ask separates emotive utterances from nonemotive utterances and determines the specific emotion types in the emotive utterances. This system is able to specify 10 distinctive emotion types as defined by Nakamura [10]. These are: joy/delight, anger, excitement, sadness/gloom, liking/fondness, fear, relief, dislike, surprise/amazement, and shyness. Our emoticon recommendation method utilizes the result of the emotion types obtained from ML-Ask and reorders the emoticon database.

The values shown in Fig. 3 are averages of ratings given by 60 Japanese university students from the previous work presented in [11]. Students were asked to rate 59 emoticons according to each of 10 distinctive emotion types on a 5-point scale. Figure 4 shows an example rating. From the total ratings, we found that 35 out of 59 emoticons express plural emotion types. Figure 5 shows the number of emoticons

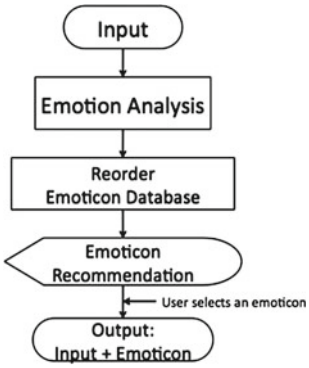

Fig. 1 System procedure

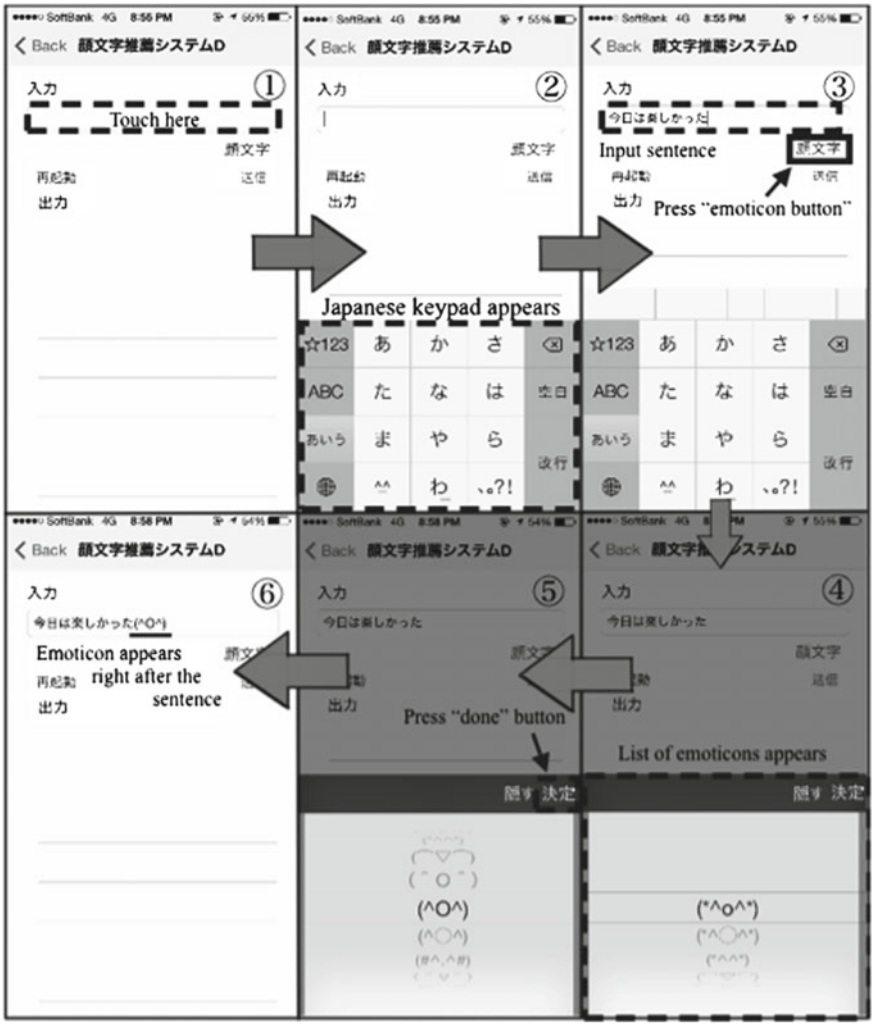

Fig. 2 Application procedure (Device: iPhone 5S, iOS 7.0.4). 1. Touch the *squared* area (①). 2. Japanese keyboard appears (②). 3. Input sentence and press "emoticon button" (③). 4. List of emoticons appears (④). 5. After choosing an emoticon, press the "done" button (⑤). 6. The emoticon is inserted right after the sentence (⑥)

that scored more than 3.0 for each of the 10 emotion types. As can be seen in Fig. 5, the number of emoticons expressing positive emotions (joy/delight, fondness/liking, and relief) was much more than other emotion types. From this result, we can assume that there are many more symbols and letters which can be used to create positive facial expressions than negative ones.

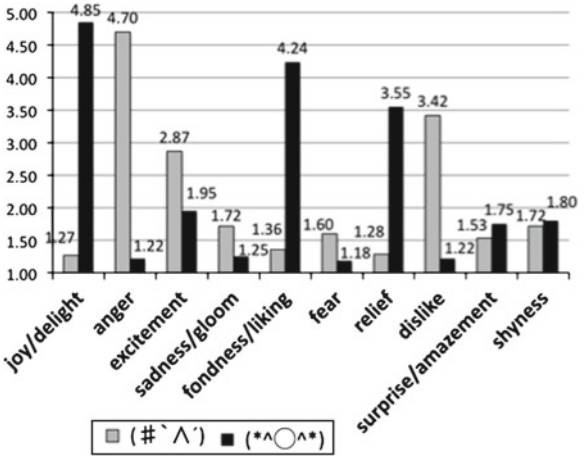

Fig. 3 Example of average rated values for two emoticons. Bars colored in *gray* and *black* are an average of rating of (#‘ ∧ ’) and (* ^ ◯ ^ *), respectively

Fig. 4 Example of emoticon ratings in each of the 10 emotions

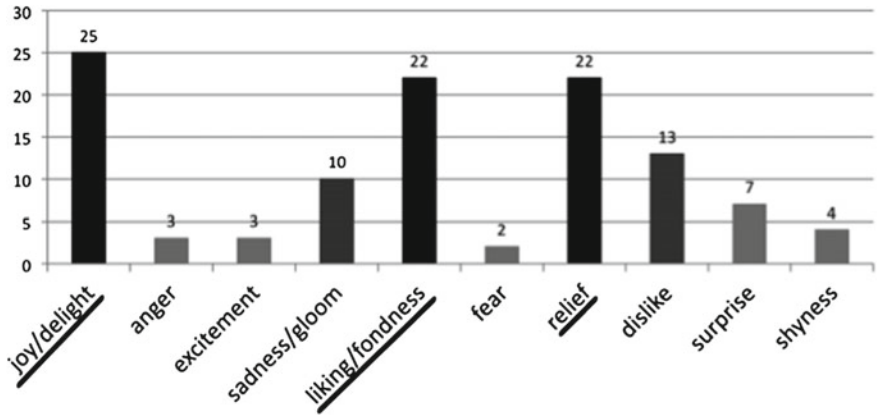

Fig. 5 Number of emoticons rated more than 3.0 for each emotion type

3.2 Integrating Proposed and Baseline Methods

The baseline method and the proposed method have their own advantages in recommending emoticons to users. The baseline method, currently used in the Japanese keypad in iOS, is useful in recommending emoticons that users choose frequently. On the other hand, the proposed method is capable of recommending emoticons according to the emotion type of the content. Therefore, we integrated the proposed and the baseline methods to use the benefits of both. The process of our integrated system is as follows: first, the system utilizes ML-Ask to analyze the emotion type of the input. Then, it sorts the selection frequency of the emoticons according to the emotion type estimated by ML-Ask, and then by the emoticon points for the emotion type. This system first collects emoticons that express similar emotions based on the input and especially considers users' emoticon preferences, so we anticipate that it may be a more user-friendly system than the two aforementioned systems.

4 Determining the Optimal Emoticon Recommendation Method

We compared the proposed method, the baseline method, and the integrated method. In order to exclude any differences in operation, we designed an application for the baseline method and the integrated method with the same operation as the proposed method (Fig. 2). These applications are usable on the iPhone (from iOS 7.0 to the latest at the time of writing). The device we used for the experiment was the iPhone 5S (iOS 7.0.4) due to its compatibility with the latest iOS at the time of writing. The experiment was carried out over 8 days from October 31, 2013 to November 8, 2013 with the cooperation of 30 Japanese undergraduate and master's students. We investigated the efficiency and user impressions of each system from (a) the ratio of emoticons chosen among the top five recommendations, (b) evaluation using the semantic differential (SD) scale and factor analysis, and by (c) asking the participants to rank the three systems based on the systems' performance and the participant's preferences for each system.

4.1 Semantic Differential Scale

The semantic differential (SD) scale was designed by Osgood et al. [12] in order to investigate user attitudes toward an object (e.g., a system, place, etc.). Briefly, the SD scale utilizes a number of scales consisting of polar opposite words such as "good–bad," "strong–weak," and "active–passive" to differentiate the meaning of concepts. Our experiment employed the SD scale with 22 bipolar words (Table 1) and the subjects' perceptions quantified on a 7-point scale. We determined the bipolar words based on our past research [11].

Table 1 22 bipolar word pairs

22 image-word pairs (translated from Japanese used in experiment)
Boring–Fun, Not impressive–Impressive, Unfriendly–Friendly
Difficult to use–Easy to use, Slow–Fast, Inconvenient–Convenient
Unnecessary–Necessary, Heavy–Light, Obscure–Clear, Dislike–Like
Old–New, Complicated–Simple, Not interested–Interested
Common–Noble, Inaccurate–Accurate, Useless–Useful
Difficult to see–Easy to see, Difficult–Easy, Difficult to choose–Easy to choose
Ordinary–Special, Dumb–Smart, Unsatisfied–Satisfied

4.2 Evaluation Experiment

4.2.1 Participants

The experiment was undertaken with the cooperation of 30 students (undergraduates and graduates). The group consisted each of 15 men and women. Their average age was 22.4 years (SD = 1.8). Among the 30 participants, 60.0 % of the students possessed an iPhone or iPad, 33.3 % possessed an Android device, and the rest possessed feature phones. Moreover, 86.7 % of the students reported that they "very often" or "somewhat often" send emails daily, and 90.0 % use emoticons "very often" or "somewhat often" when sending email.

4.2.2 Procedure

The procedure of the experiment was as follows:

1. Participants first filled out basic information (their university year group, sex, age, faculty, whether they possess a smartphone, whether they send emails daily, and whether they use emoticons in sending messages daily).
2. Participants tested one of the three systems. The order in which a participant tested the three systems was decided by random selection in order to examine the difference between participants using each of the systems at the beginning.
3. Participants rated the system using 22 bipolar words on a 7-point scale (Table 1).
4. Participants tested the other two systems as written above in Steps 2 and 3.

The contents of the input were decided in advance. We prepared a list of 15 sentences that each included one emotive word, and showed it to the participants, asking them to enter each sentence in each of the three systems. The sentences for the list were selected from participants' inputs from a previous experiment [11]. These were typed by the participants on the sole condition of using only one emotive word in each sentence. We performed a preliminary experiment to examine how strongly the chosen sentences express one of the 10 emotion types by asking 10 Japanese subjects

Table 2 Example of sentences shown to participants

Japanese sentence	Transliteration	Translation	Emotion
その漫画は好きだよ。	*Sono manga wa suki desu yo*	I like this comic book	Liking/fondness (4.9) (positive)
それはちょっと恥ずかしい。	*Sore wa chotto hazukashi*	This is a little embarrassing	Shyness (4.3) (neutral)
怯えてしまう。	*Obiete shimau*	I am frightened	Fear (4.8) (negative)

to rate them on a 5-point scale (minimum: 1.0, maximum: 5.0; average points collected from respondents are written after the emotion types in Table 2). The list was comprised of three five-sentence groups, each group expressing one of the positive emotions, a random selection from joy/delight, relief, and liking/fondness, one of the neutral emotions, a random selection from surprise/amazement, excitement, and shyness, and one of the negative emotions, a random selection from fear, sadness/gloom, anger, and dislike (examples shown in Table 2).

5 Results and Discussions

5.1 Proportion of Emoticons Chosen Among the Top Five from Each System

Table 3 shows the results of the proportion of emoticons chosen among the top five recommended by each system. Our proposed system scored 57.6 % and our integrated (baseline + proposed) system scored the highest at 73.0 %, both of which are major improvements over the baseline system. From these results, we can assert that recommending emoticons depending on the emotion type of the input is effective for users. Also, when we examined users' chosen emoticons, it seemed that users have their own emoticon preferences for each emotion type; therefore, the performance improves when we integrate users' past selection data (baseline method) with the emotion-based recommending method (proposed method).

We broke down the overall results into positive (joy/delight, liking/fondness, and relief), negative (sadness/gloom, anger, fear, and dislike), and neutral (surprise/amazement, excitement, and shyness) to investigate whether there is a difference in choosing emoticons by the valence of the input (Tables 4, 5 and 6). We discovered that the results of the baseline method for the negative statements (Table 5) were a little lower than that of positive and neutral statements. This is due to the fact that negative emoticons were placed lower in order at the very beginning so users had to scroll down to find emoticons. This can also be said for the emoticon database in the

iOS Japanese keypad, that is, many positive emoticons are placed at the top, whereas negative emoticons are arranged in the lower part in the database. Therefore, replacing emoticon recommendation depending on the valence of the input is necessary in order to improve the quality of the performance. We also determined that our integrated system performs slightly better for negative statements (Table 5) than other statement types. This result comes from the smaller number of negative emoticons than that of positive emoticons in the database. The number of emoticons for surprise/amazement, shyness, and excitement in the database was also smaller than that of positive emoticons; however, it did not give a result (Table 6) as high as that of negative statements, because most of these emotions also imply either positive or negative contexts in the statement (e.g., "She was thrilled to death to get the flowers" (excitement and joy/delight), "I was shocked to see a ghost" (fear and surprise/amazement), etc.). Therefore, we should consider whether the statement is weighted toward positive or negative when the statement contains these three emotion types.

5.2 Participants' Attitudes from SD Scale

Next, we collected and calculated the average of respondents' attitudes toward each of the three systems using an SD scale (Fig. 6). In Fig. 6, numbers closer to one have strong impressions of the words on the left, whereas numbers closer to seven are better characterized by the words on the right. The averages are shown under each system.

From the results shown in Fig. 6, we discovered that our integrated system scored the highest among the three systems for 15 word pairs out of 22 word pairs. The overall average of the integrated system was 5.4 points, which was slightly higher than the proposed system (5.3 points). The baseline system scored 4.1 points, therefore, we verified that methods recommending emoticons according to emotion types from input are more effective than the baseline method. We also found that our integrated system (4.9 points) and our proposed system (5.4 points) scored lower than the baseline system (5.6 points) for the word pair "complicated–simple." We assume that most participants rated this by considering the process of the system recommending emoticons to them.

Table 3 Proportion of emotions chosen among the top five recommendations

	Baseline (%)	Proposed (%)	Integrated (baseline + proposed) (%)
Overall	29.5	57.6	**73.0**
Men	26.0	59.5	**74.9**
Women	33.3	55.6	**71.0**

Table 4 Proportion of emotions chosen among the top five recommendations (Positive)

	Baseline (%)	Proposed (%)	Integrated (baseline + proposed) (%)
Overall	32.2	57.5	**71.6**
Men	30.1	60.6	**75.3**
Women	34.2	54.8	**68.0**

Table 5 Proportion of emotions chosen among the top five recommendations (Negative)

	Baseline (%)	Proposed (%)	Integrated (baseline + proposed) (%)
Overall	23.9	67.4	**76.7**
Men	17.1	65.3	**76.4**
Women	31.7	69.4	**77.0**

Table 6 Proportion of emotions chosen among the top five recommendations (Neutral)

	Baseline (%)	Proposed (%)	Integrated (baseline + proposed) (%)
Overall	32.9	50.0	**71.1**
Men	31.5	56.2	**73.2**
Women	34.2	44.0	**69.0**

5.2.1 Factor Analysis of the SD Scale Ratings

We carried out a factor analysis of the SD scale ratings in order to condense a large number of variables into a few interpretable underlying factors and summarize the respondents' perception toward each of the three systems. The factor analysis resulted in three factors with eigenvalues exceeding 1.0 which accounted for 66.4 % of the variance. Table 7 shows the varimax rotation factor loadings for the 22-bipolar word pairs.

The first factor is made up of 16 scales and can be described as "users' impression of the system" (e.g., whether they feel the system is difficult or easy to use, whether they are satisfied with the system, etc.). The second factor is made up of three scales (common–noble, ordinary–special, and old–new). These word pairs can be summarized as "novelty of the system." The third factor was also comprised of three factors (slow–fast, heavy–light, and complicated–simple); therefore, we named this factor "system performance."

We plotted the 22 bipolar word pairs with groups of respondents categorized by system and gender (Figs. 7 and 8). As shown in Fig. 7, we discovered that our integrated system ("I" in Fig. 7) demonstrated the highest novelty and the most positive impression among the three systems, whereas the baseline system ("C" in Fig. 7) was ranked by far the lowest in both these aspects. Our proposed system ("C" in Fig. 7) also produced a positive impression similar to our integrated system, and slightly positive in terms of system novelty. When we consider the difference between

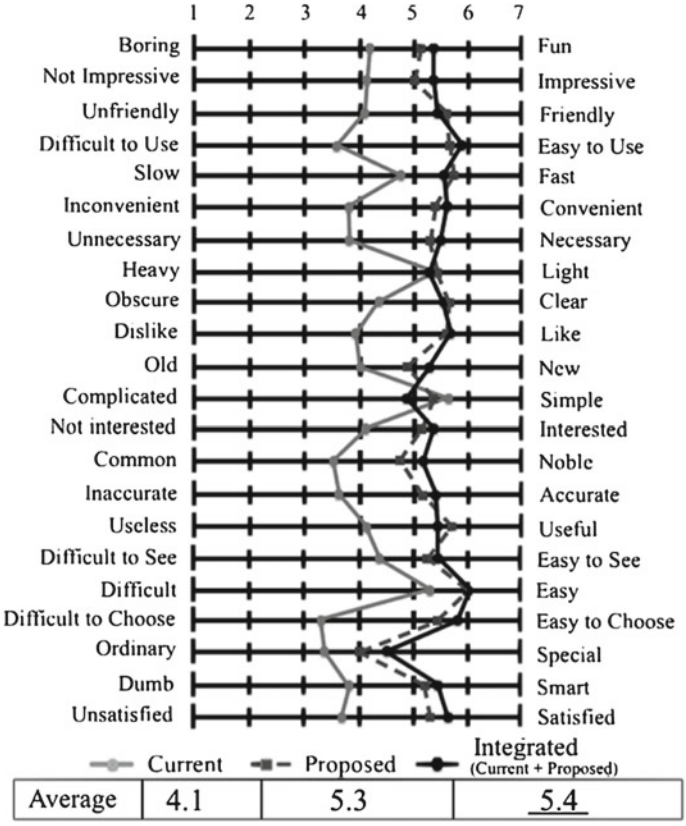

Fig. 6 Results of evaluation using SD scale

genders, it is apparent that the male users have the most positive perceptions toward the integrated system ("M" circled with "I" in Fig. 7) among the three systems, while the female users seemed to like the proposed system ("F" circled with "P" in Fig. 7) the best, however, the female users reported their highest impression of novelty ("F" circled with "I" in Fig. 7) for the integrated system. For the third factor, "system performance," we discovered that the users felt that our proposed system ("P" in Fig. 8) seemed to perform the fastest and the lightest of all systems. We also compared the perceptions of system performance according to gender and found that the female users felt that the proposed system ("F" circled with "P" in Fig. 8) performs the best, while the male users preferred the baseline system ("M" circled with "C" in Fig. 8). Our integrated system produced a relatively lower impression ("I" in Fig. 8) for this factor, probably due to the complexity of the method of recommending emoticons compared to the proposed and the baseline methods.

Table 7 Factor Loadings of each of the 22 bipolar word pairs in the SD scale ($\geqq 0.3$)

22-Bipolar word pairs (Name given to pair)	Factor 1 (Impression of the system)	Factor 2 (Novelty of the system)	Factor 3 (System performance)
Difficult to use–Easy to use (ETU)	0.88		
Unsatisfied–Satisfied (SAT)	0.88	0.32	
Inconvenient–Convenient (CON)	0.86		
Unnecessary–Necessary (NEC)	0.82		
Difficult to choose–Easy to Choose (ETC)	0.82		
Dislike–Like (LIK)	0.82		
Useless–Useful (USE)	0.80		
Unfriendly–Friendly (FRI)	0.72	0.32	
Dumb–Smart (SMA)	0.72	0.50	
Inaccurate–Accurate (ACC)	0.71		
Obscure–Clear (CLE)	0.67		0.40
Not interested–Interested (INT)	0.64	0.54	
Difficult to see–Easy to see (ETS)	0.63		
Not impressive–Impressive (IMP)	0.60	0.51	
Boring–Fun (FUN)	0.58	0.42	
Difficult–Easy (EASY)	0.53		0.39
Common–Noble (NOB)	0.38	0.80	
Ordinary–Special (SPE)		0.72	
Old–New (NEW)	0.44	0.71	
Slow–Fast (FAS)			0.75
Heavy–Light (LIG)			0.72
Complicated–Simple (SIM)			0.31
Eigenvalues	12.6	1.8	1.3
% of total cumulative variance	41.5	57.7	66.4

5.3 Rankings Based on the Systems' Performance and Users' Preference

We also asked the respondents to rank the three systems based on performance and which of the three systems they prefer. Tables 8 and 9 show the results of this ranking. As shown in Table 8, 23 out of 30 participants ranked our integrated system

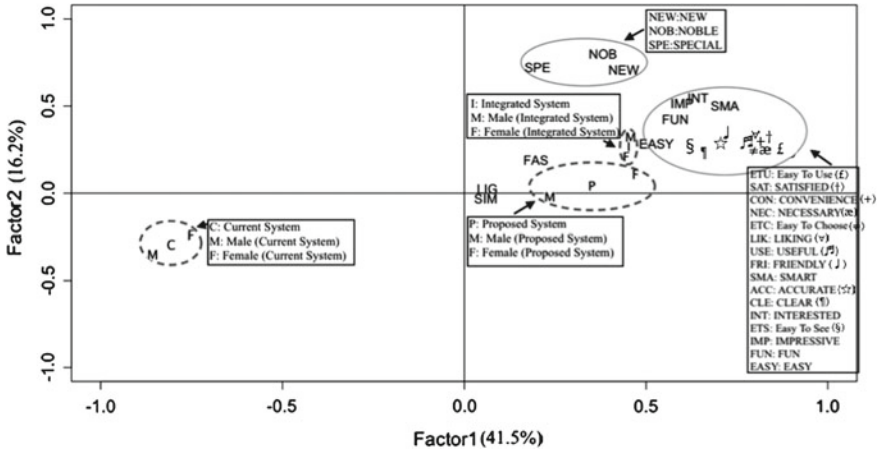

Fig. 7 Biplot of the two factor models for Factor 1 and 2. X-axis is Factor 1 (Factor 1 explains 41.5 % of the total variance), y-axis is Factor 2 (Factor 2 explains 16.2 % of the total variance)

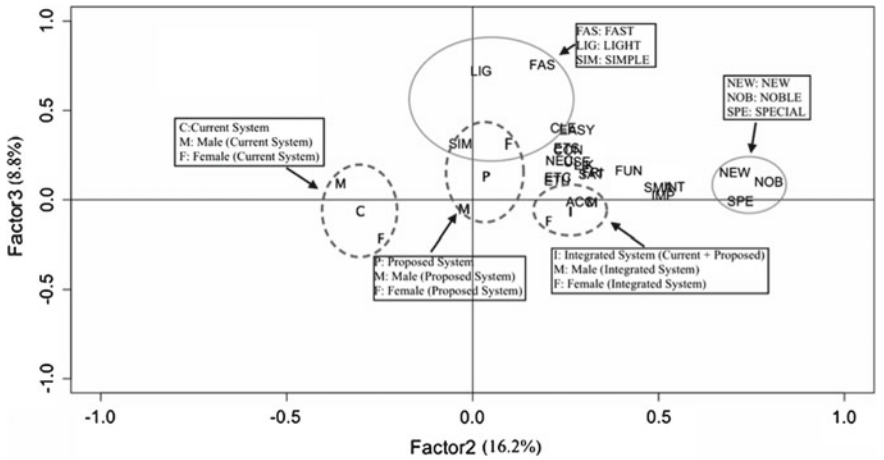

Fig. 8 Biplot of the two factor models for Factor 2 and 3. X-axis is Factor 2 (Factor 2 explains 16.2 % of the total variance), y-axis is Factor 3 (Factor 3 explains 8.8 % of the total variance)

as performing the best, 16 out of 30 participants ranked the proposed system as second, and 21 out of 30 participants ranked the baseline system as third. As shown in Table 9, the ranking was in descending order of: our integrated system (21 out of 30 participants), our proposed system (14 out of 30 participants), and the baseline system (19 out of 30 participants). From these results, we concluded that our integrated system achieved a great improvement over the baseline system in terms of system performance and user preferences.

Table 8 Ranking based on the systems' performance (numbers are the total of people)

System	1st	2nd	3rd
Integrated	**23**	6	1
Proposed	6	**16**	8
Baseline	1	8	**21**

Table 9 Proportion of emotions chosen among the top five recommendations (Neutral)

System	1st	2nd	3rd
Integrated	**21**	8	1
Proposed	6	**14**	10
Baseline	3	8	**19**

6 Conclusions

In this paper, we presented two emoticon recommendation methods based on users' past emoticon selection and emotional statements contained in the input. The main procedures of these two methods share the same process of analyzing emotions from user-entered sentences by using the affect analysis system ML-Ask, but differ in their methods of reordering the emoticon database and recommending appropriate emoticons to users. Our originally created database utilized an idea by Kawakami [9], and comprised of 59 emoticons with the points expressed from each of 10 distinctive emotions.

Evaluation experiments were performed to compare the performance of the three systems. We discovered that approximately 73.0 and 57.6 % of chosen emoticons were among the top five recommendations by our integrated system (the incorporation of the baseline and the proposed systems) and our proposed system, respectively. On the other hand, the baseline system used in the Japanese iPhone keypad only scored 29.5 % in the same experiment. We also confirmed that our integrated and proposed systems scored 5.4 points and 5.3 points, respectively, in evaluation using a semantic differential scale, which was relatively larger than the baseline system of 4.1 points. Furthermore, the results of a factor analysis demonstrated that users perceived the highest novelty and had the most positive impression towards our integrated system, whereas the baseline system was rated the lowest in these factors. The overall ranking of the three systems was in descending order of: our integrated system, our proposed system, and the baseline system, in terms of system performance and users' preferences. From the overall results, we confirmed that emotion plays a major role when recommending appropriate emoticons to users. Furthermore, users have their own preferences when selecting emoticons with their input, therefore, the integrated method is the most user-friendly.

We believe that we can expect further improvements in recommending more appropriate emoticons to users. First of all, in future work, we could recommend more

suitable emoticons for inputs expressing neutral emotion types (surprise/amazement, shyness, and excitement) by analyzing whether the input is weighted toward either positive or negative. For example, a sentence like "She was thrilled to death to get the flowers" expresses both excitement and joy/delight and so is weighted toward a positive statement, however, a sentence like "I was shocked to see a ghost" expresses both surprise/amazement and fear, and so is weighted to a negative statement. Second, we intend to apply an existing machine learning method to learn which kinds of emoticons are preferred for which words in the sentence, so that our system will also work with sentences with no emotive words. Lastly, expansion of the emoticon database is necessary in order to allow larger numbers of emoticons to be inserted easily. Also, more emoticons in the database will be helpful for discovering the types of symbols that articulate each emotion type, and create a system to automatically generate emoticons suitable to the input.

The emoticon recommendation system is not only useful for assisting users to choose an appropriate emoticon for Japanese messages, but also can be utilized in various ways. First, the system can be utilized for any language, though the emoticon database may need a little adjustment to the emotional strength value due to the difference in interpreting emoticons across cultures [2]. Second, our approach is also capable of working with pictograms that are input along with characters using mobile phones. Third, it is possible to use our system with a text-based dialogue system in order to express the feeling using emoticons and show friendliness toward the interlocutor.

References

1. Mehrabian A (1971) Silent messages, 1st edn. Wadsworth, Belmont
2. Park J, Barash V, Fink C, Cha M (2013) Emoticon style: Interpreting differences in emoticons across cultures. In: The 7th international AAAI conference on weblogs and social media. AAAI Press, Palo Alto, pp 466–475
3. Derks D, Bos AER, Grumbkow JB (2008) Emoticons in computer-mediated communication: social motives and social context. In: Cyberpsychology & Behavior, vol 11(1). Mary Ann Liebert, New York, pp 99–101
4. Lo SK (2008) The nonverbal communication functions of emoticons in computer-mediated communication. In: Cyberpsychology & Behavior, vol 11(5). Mary Ann Liebert, New York, pp 595–597
5. Kato S, Kato Y, Scott D (2009) Relationships between emotional states and emoticons in mobile phone email communication in Japan. In: International journal on e-learning, vol 8(3). AACE Press, Waynesville, pp 385–401
6. Ptaszynski M, Dybala P, Rzepka R, Araki K (2008) Effective analysis of emotiveness in utterances based on features of lexical and non-lexical layer of speech. In: The 14th annual meeting of the association for natural language processing, pp 171–174
7. Ptaszynski M, Maciejewski J, Dybala P, Rzepka R, Araki K (2010) CAO: A fully automatic emoticon analysis system based on theory of kinesics. IEEE Trans Affect Comput 1:46–59
8. Emura Y, Seki Y (2012) Facemark recommendation based on emotion, communication, and motion type estimation in text. IPSJ Sig Notes 85:1–7
9. Kawakami M (2008) The database of 31 Japanese emoticon with their emotions and emphasis. In: The human science research bulletin of Osaka Shoin Women's University, vol 7, pp 67–82

10. Nakamura A (1993) Dictionary of emotive expressions (Kanjyo Hyogen Jiten) (in Japanese). Tokyodo Publishing, Tokyo
11. Urabe Y, Rzepka R, Araki K (2013) Emoticon recommendation for Japanese computer-mediated communication. In: The proceedings of 2013 IEEE seventh international conference on semantic computing, Irvine, pp 25–31
12. Osgood CE, Suci GJ, Tannenbaum PH (1957) The measurement of meaning. University of Illinois Press, Urbana

Accuracy Versus Novelty and Diversity in Recommender Systems: A Nonuniform Random Walk Approach

Georgios Alexandridis, Georgios Siolas and Andreas Stafylopatis

Abstract In this chapter, we focus on recommender systems that are enhanced with social information in the form of trust statements between their users. The trust information may be processed in a number of ways, including the random walks in the social graph, where every step in the walk is chosen almost uniformly at random from the available choices. Although this strategy yields satisfactory results in terms of the novelty and the diversity of the produced recommendations, it exhibits poor accuracy because it does not fully exploit the similarity information among users and items. Our work tries to model user-to-user and user-to-item relation as a probability distribution using a novel approach based on Rejection Sampling in order to decide its next step (biased random walk). Some initial results on reference datasets indicate that a satisfying trade-off among accuracy, novelty, and diversity is achieved.

Keywords Recommender systems · Trust networks · Non-uniform random walks · Rejection sampling · Accuracy · Novelty · Diversity

1 Introduction

It is a fact that the emergence of Online Social Networks (OSN) has altered our everyday experience with the Internet and the World Wide Web. A number of new application domains have been born, while others, traditional ones, have been enriched. The latter is the case with the recommender systems (RS), where OSN have leveraged user experience by allowing a more thorough interaction that surpasses the traditional user-to-item review.

G. Alexandridis (✉) · G. Siolas · A. Stafylopatis
School of Electrical and Computer Engineering,
National Technical University of Athens, 157 80 Zografou, Athens, Greece
e-mail: gealexandri@islab.ntua.gr

G. Siolas
e-mail: gsiolas@islab.ntua.gr

A. Stafylopatis
e-mail: andreas@cs.ntua.gr

Ö. Ulusoy et al. (eds.), *Recommendation and Search in Social Networks*,
Lecture Notes in Social Networks, DOI 10.1007/978-3-319-14379-8_3

Indeed, research in the traditional RS field had come to a relative standstill prior to the advent of social networks. Although the application of the latter into the former is still novice, current state-of-the-art research in the field involves the integration of OSN in one or another form [1, 2]. It could be further argued that the blending of the two areas has brought about a new research field, that of the socially aware recommendation.

Social recommender systems (SRS) model and exploit user-to-item and user-to-user interaction in a plethora of ways [3, 4]. The addition of social information generally leads to more novel and diverse recommendations. However, this does not necessarily imply that the recommendations would be accurate altogether; indeed SRS have to be selective in the volume of information they incorporate. In this context, random walks on the social graph are fit for this purpose, since they focus on those subsets of the data that they find useful. For this reason, almost from the very beginning, they have been a natural choice for researchers in the field and they have been used in the implementation of widely used and successful systems [5, 6]. It is not only our belief, but also that of the community [7, 8] that random walks have not yet revealed their full potential and that there is still room both for improvements in existing algorithms and the exploitation of other aspects of the random walks that are currently unexplored. In continuation of a preliminary work [9], this work tries to exploit the random walks from a different viewpoint; that of bridging the gap between recommendation accuracy on the one hand and novelty and diversity on the other.

2 Social Recommender Systems Based on Trust

Traditional RS can be extended by incorporating the interaction among users into them. This interaction may take place in a number of ways, the most common of which is Trust. It is the most simple form of user relation, where a user expresses his opinion on another user's behavior. Trust statements could either be binary (i.e., trust/distrust) or they may range over a broader set of values (usually in the [0, 1] interval). It should be noted that trust does not necessarily imply correlation in the rating behavior [10].

The public release of socially enhanced recommendation datasets, such as the Filmtrust or the Epinions datasets (Table 1) has spurred interest in SRS. Since most of these datasets disclose trust information among their users, a substantial amount of the work in the area has evolved around trust-aware RS.

2.1 Trust Aggregation

A common way of processing the trust information of SRS is by aggregation; that is, to try to build a metric that would accumulate the available trust statements in the system. An obvious choice would be to consider all the paths that end up to a

Table 1 Recommender datasets used in the experiments

	Filmtrust	Epinions
Users	1919	49290
Items	2018	139738
Ratings	33526	664824
Ratings' density	1.15 %	0.01 %
Trust statements	1591	487182
Global clustering coefficient	0.0004	0.0002

particular user, in an effort to estimate his or her importance. Such SRS are also called *Reputation Systems* and their operation bears resemblance to the way the *PageRank* scoring algorithm works [7]. Although important research has been conducted in this direction, global trust metrics are not particularly suitable for the recommendation task. The main reason is that recommendations have to be personalized and in that sense the reputation of each user could not be constant; it depends on the viewpoint of each other user.

Local trust metrics, on the other hand, put the emphasis on each individual user and depart from him/her in order to explore the network. One of the earliest works in the field include the *gradual trust metric MoleTrust* [10] proposed by Massa and Avesani. The graph is first transformed into an acyclic form (a tree) by removing all loops in it and then the trust statements are accumulated in a depth-first fashion, starting from each user, up to each and every other user in the network. The *propagation horizon* determines the length of the exploration; the most common forms being MoleTrust-1, where only the users that target user trusts are considered, and MoleTrust-2, where the exploration also includes those trusted by those the target user trusts. If T_{u_t} is the set that includes all users in u_t's network that have rated item i_t (which has not been evaluated by u_t yet), then the recommendation value $\widehat{r_{u_t,i_t}}$ is approximated using the following formula (*trust-based collaborative filtering*) :

$$\widehat{r_{u_t,i_t}} = \overline{r_{u_t}} + \frac{\sum\limits_{u \in T_{u_t}} t_{u_t,u}(r_{u,i_t} - \overline{r_u})}{\sum\limits_{u \in T_{u_t}} t_{u_t,u}} \tag{1}$$

where $\overline{r_{u_t}}$ is the mean of the ratings u_t has provided so far and $t_{u_t,u}$ the amount of trust u_t places on u.

Another popular gradual trust metric, proposed by Golbeck, is *TidalTrust* [6]. TidalTrust is different from MoleTrust in the sense that no propagation horizon is required for the accumulation of trust; instead the shortest path from the target user to each other user in the network is computed. All paths above a predefined threshold form the *Web of Trust* (*WOT*) for that particular user. If there exists more than one path between two users, then the one with the biggest value is chosen. If WOT_{u_t} is the set that includes those users in u_t's web of trust network that have rated item i_t, then the recommendation value $\widehat{r_{u_t,i_t}}$ is approximated using the formula (*trust-based*

weighted mean):

$$\widehat{r_{u_t,i_t}} = \frac{\sum\limits_{u \in T_{u_t}} t_{u_t,u} r_{u,i_t}}{\sum\limits_{u \in T_{u_t}} t_{u_t,u}} \tag{2}$$

2.2 Random Walks

Trust aggregation approaches, however, are impractical in the case of a large OSN, where a user's friends, friends-of-friends, etc., could quickly scale to a magnitude of thousands. For this reason, random walks have become a natural choice for researches in the field of SRS [3, 8]. One of the first works on the subject is the *TrustWalker* system [1] which performs simple random walks on the trust graph, by defining transition and exit probabilities at each step of the walk. Neighbors, however, need not be chosen uniformly at random; in [2], the graph is initially traversed looking for the existence of strongly connected components. Then a nonuniform random walk is performed whose restarting probability depends on whether the currently active node is a member of a strongly connected component or not.

Random walks in the connected components of the graph assume the properties of *Markov Chains* (steady-state distribution, irreducibility, etc.). These properties have been further exploited by researchers as in [4], where a semi-supervised classification algorithm is applied in order to produce recommendations. The algorithm estimates the probability of a random walk starting at item y to terminate at the target user and these probabilities are considered to be markovian variables.

3 Design Aspects and Motivation

Although random walks in trust networks have been studied thoroughly, we believe there is still room for improvement. We must depart from the simple random walks that select their next step uniformly (or almost-uniformly) at random and introduce some bias toward "better" nodes. That is, we should discriminate our neighbors by increasing the transition probability toward similar users (defined in a recommendation context) and at the same time decreasing the transition probability toward dissimilar users.

3.1 Measuring Correlation

Unfortunately, trust and similarity are two concepts that do not necessarily coincide in SRS [10]. In the recommendation domain, two users are considered to be correlated

(similar) if they rate the same items in the "same" fashion. A number of metrics, derived from the statistical literature, measure how close two populations, i.e., U_x and U_y, are. U_x and U_y could be the ratings of user $u_x \in U$ and $u_y \in U$ on the same set of items I.

Statistical correlation has been extensively analyzed in the RS context and it has been found out that one of the most satisfactory metrics of correlation is the *Pearson Correlation Coefficient* [11], especially when the sets U_x and U_y coincide to a large extent. Unfortunately, this is not always the case in RS, particularly in sparse datasets. In such cases, other metrics like the *Log Likelihood* similarity or the *City-Block* (*Manhattan*) similarity yield better results.

3.2 Performing the Random Walk

Theoretically, better recommendations can be achieved if we walk toward more similar users (compared to selecting them uniformly at random). To further elaborate on this idea, we might consider the similarity metrics between a user and its direct neighbors in the trust network as samples of an unknown probability distribution that measures how close two neighbors actually are in their rating behavior. By moving toward like-minded neighbors (and not like-minded users as is the case with collaborative filtering), we increase our chances of getting a correct recommendation.

An obvious choice would be to pick the most similar neighbor each time. However, this is not the best strategy mainly because the ratings are not evenly distributed over all users and items in the dataset but follow a *Zipf Law* instead; a few users (items) issue a lot of ratings while most users (items) have only issued a few, belonging in the *long-tail* of the distribution. A deterministic algorithm would always pick the small slice of users and items with the most ratings and would consequently make recommendations from a restricted set of users, thus having a negative effect on the novelty and serendipity of the proposed items. Clearly, probabilistic algorithms allow for better exploration of the available choices contributing to the overall serendipity of the recommendation process.

The last issue that remains to be resolved is the fact that the target distribution we would like to sample from still remains unknown. For this reason, we first turn the similarity metrics into probabilities (by dividing each one with their sum) and then use an acceptance/rejection sampling algorithm to generate samples from.

4 The Biased Random Walk Algorithm

Our proposed random walk algorithm works in three phases. In the first phase and for each user, it retrieves from the user database all those users that have at least one rating in common with him/her (forming the set of the *Correlated Neighbors C*) and all those users that are trusted by the target user (forming the set of the *Trusted*

Neighbors T). Contrary to what might have been expected, these two sets are to a very large extent not overlapping. Therefore, a decision has to be made on which set of users to follow. A natural strategy would be to sample from each set based on its relative importance. That is, with a probability $P_T = \frac{|T|}{|C|+|T|}$ the next user in the walk is selected from T and with a probability of $P_C = 1 - P_T = \frac{|C|}{|C|+|T|}$ from C. In the first case, the next user is selected uniformly at random from T since the trust statements are binary. However, this rule does not hold for the second case, as users are correlated to one another to a different degree. It is this point where rejection sampling reveals its potential.

4.1 Rejection Sampling

The concept behind *Rejection Sampling* (or *acceptance-rejection* algorithm) is to use an easy-to-sample probability distribution $\mathscr{G}(x)$ as an instrument to sample from the unknown distribution $\mathscr{F}(x)$. $\mathscr{G}(x)$ is also referred to as the *proposal distribution*. Let $f(x)$, $g(x)$ be the respective probability distribution functions. The only prerequisite of this method is that the support of $g(x)$ dominates the support of $f(x)$ up to a proportionality constant c. That is, the following inequality must hold true:

$$f(x) \leq cg(x), c < \infty, \forall x \in \mathscr{X} \tag{3}$$

where $\mathscr{X}$ denotes the sample space.

Next, a number n is drawn uniformly at random from $\mathscr{U}(0, 1)$ along with a sample $x_i \in \mathscr{X}$ according to $\mathscr{G}(x)$ ($x_i \sim \mathscr{G}(x)$). Then the inequality $n < \frac{f(x_i)}{cg(x_i)}$ is checked for its validity; if it holds, x_i is considered to be a valid sample drawn from $f(x)$, otherwise it is rejected and new samples n, x_i are drawn from the respective distributions.

Our recommendation algorithm performs a *Biased Random Walk* by applying the rejection sampling algorithm described earlier in order to decide its next step. For this reason, it is called *Biased RW-RS*. The target probability distribution $f(x)$ is constructed by dividing the similarity between the target user and each of its similar neighbors with the sum of their similarities. The uniform distribution $\mathscr{U}(x)$ is used as the proposal distribution and c is approximated by ensuring that the inequality $f(x) < cu(x)$ holds at each point. We then proceed to the rejection sampling method described in the function *RejectionSampling* (Fig. 1)

4.2 Terminating the Walk

An important decision to be made is when to stop the random walk. Stopping the walk early prevents the RS from exploring the user and item space, while stopping the

Require: Target User u_t
$u_c \leftarrow u_t$
while the walk is not terminated **do**
  $(C,T) \leftarrow$ SPLITNEIGHBORS(u_c)
  Sample $n \sim \mathcal{U}(0,1)$
  if $n \leq \frac{|T|}{|C|+|T|}$ **then** ▷ u_c is the current node of the walk
    Sample u_c uniformly at random from T
  else
    Sample $u_c \sim$ REJECTIONSAMPLING(T, u_c)
  end if
end while
return Visited nodes during the walk ▷ How often they were accessed

function REJECTIONSAMPLING(S, u_c)
  $sum \leftarrow 0$
  for $u_n \in Neighbors(u_c)$ **do**
    $s_{u_c,u_n} \leftarrow$ SIMILARITY(u_c, u_n)
    $sum \leftarrow sum + s_{u_c,u_n}$
  end for
  $\mathcal{G}(x) \leftarrow \mathcal{U}(\min s_{u_c,u_n}, \max s_{u_c,u_n})$ ▷ The proposal distribution becomes the uniform distribution defined over the space between the smallest and the largest similarity value
  $c \leftarrow 0$
  for $u_n \in Neighbors(u_c)$ **do** ▷ Turn Similarity into a Probability Distribution Function
    $f(u_n) \leftarrow \frac{s_{u_c,u_n}}{sum}$
    if $c < \frac{f(u_n)}{g(u_n)}$ **then**
      $c \leftarrow \frac{f(u_n)}{g(u_n)}$
    end if
  end for
  repeat
    Sample $n \sim \mathcal{U}(0,1)$
    Sample $x_i \sim \mathcal{G}(x)$
  until $n < \frac{f(x_i)}{c.g(x_i)}$
  return x_i
end function

Fig. 1 Biased Random Walk Rejection Sampling Algorithm

walk late has the risk of ending up in regions too far away from the target user. Since in most SRS the ratings' density is sparse and the global clustering coefficient of the social graph is very small, a simple probabilistic criterion is employed: with a fixed probability P_c (at each step), the walk continues and with probability $P_t = 1 - P_c$ the walk terminates (Bernoulli trial). The most widely adopted value for the termination probability is attributed to the PageRank algorithm [2] and is set at $P_t = 0.15$. After the walk termination, user nodes are ranked according to their relevance to the target user (how often they were visited during the walk) and recommendations are produced out of the most relevant ones.

5 Experiments

We have evaluated the performance of the *Biased RW-RS* algorithm into two different datasets. The first one was crawled from the Filmtrust website [12] and contains 33,526 ratings given by 1,919 users on 2,018 items, along with 1,591 trust statements. The second dataset was crawled from the Epinions website [13] and contains 664,824 ratings that 49,290 users have given to 139,738 items along with 487,182 trust statements. Both datasets are extremely sparse and the corresponding trust networks are extremely weak, following a Zipf law (Sect. 3). We have also examined three different correlation metrics as a similarity measure (line 16 of Algorithm 1); Pearson Correlation, Log Likelihood, and Manhattan Distance and we came to the conclusion that the last one is more suited for the datasets at hand.

5.1 Reference Systems

In order to better estimate the performance of the *Biased RW-RS* algorithm, we are presenting a number of reference RS (both traditional and social) and we are having them evaluated on the two datasets described above.

5.1.1 Baseline Systems

The purpose of the *Baseline Systems* is to estimate the relative improvements of the other systems. The *Random* RS would simply output a (uniformly) random value as a recommendation to each user, while the *Always-Max* RS would recommend each and every item to the target user with the biggest possible value.

5.1.2 Collaborative Filtering and Content-Based Approaches

The simple content-based and collaborative filtering recommendations are produced according to the widely adopted in the recommender systems' literature Resnick's formula [14]. The predicted rating that a particular target user would have given to a specific unseen item is determined by two factors: the target user's average rating on the other items he/she has evaluated so far and the ratings on the specific item given by the other users in the dataset:

$$\widehat{r_{u_t,i_t}} = \overline{r_{u_t}} + \frac{\sum_{i=1}^{|N|} w_{u_t,u_c}(\overline{r_{u_c}} - r_{u_c,i_t})}{\sum_{i=1}^{|N|} w_{u_t,u_c}} \tag{4}$$

where u_t is the target user and $u_c \in N$ all of his neighbors with whose the similarity value w_{u_t,u_c} can be computed.

5.1.3 Trust-Based Approaches

The *trust-based approaches* refer to the respective trust aggregation methodologies outlined in Sect. 2.1 (Eqs. 1 and 2). Especially for the MoleTrust case, the numerical suffix indicates the maximum propagation horizon.

6 Evaluation Metrics

6.1 Predictive Accuracy

Traditionally, the RS performance has been evaluated against the *Predictive Accuracy Metrics*[11]. Their purpose is to measure how close the predicted value $\widehat{r_{u,i}}$ is to a retained actual rating $r_{u,i}$. For this reason, the dataset is split into disjoint parts (sets) one of which is selected as the test set while the others form the training set. In our experiments, we have used the *tenfold cross-validation* model and the results on this category of metrics (Tables 2 and 3) are averaged for the 10 runs of the model.

The most widely adopted predictive accuracy metric is the *root mean square error* (RMSE), which is defined over a test set T as:

$$RMSE = \sqrt{\frac{1}{|T|}\sum_{n=1}^{|T|}(\widehat{r_{u,i}} - r_{u,i})^2} \tag{5}$$

where $|T|$ is the cardinality of the test set.

A similar metric is the *mean absolute error* (MAE), which measures the difference between the output of the RS on a given input sample versus its expected value, averaged over all samples in T:

$$MAE = \frac{1}{|T|}\sum_{n=1}^{|T|}|\widehat{r_{u,i}} - r_{u,i}| \tag{6}$$

The two aforementioned metrics weight each prediction error the same and therefore favor users with more ratings. In order to introduce a trade-off between users with many ratings and cold-start users, Massa and Avesani [15] proposed the *mean absolute user error* which functions exactly like MAE; the only difference being that it first calculates the MAE over the ratings of a specific user and then computes the average of the MAE of all users:

Table 2 Predictive accuracy metrics: results on all users

Datasets	Filmtrust				Epinions			
Performance metrics	RMSE	MAE	MAUE	Coverage (%)	RMSE	MAE	MAUE	Coverage (%)
A. Baseline								
A.1 Random	1.53	1.25	1.26	100.00	1.94	1.61	1.63	100.00
A.2 Always-Max	1.35	1.00	0.90	100.00	1.57	1.01	0.97	100.00
B. Collaborative filtering								
(All neighbors)	0.88	0.70	0.68	93.65	1.07	0.81	0.82	79.57
C. Content-based recommendation								
(Nearest n items)	0.78	0.60	0.61	72.71	1.37	0.99	1.00	22.92
D. Trust-based approaches								
D.1 MoleTrust-1	0.97	0.73	0.74	18.64	1.23	0.91	0.95	25.58
D.2 MoleTrust-2	0.91	0.70	0.72	24.76	1.16	0.88	0.93	56.52
D.3 MoleTrust-3	0.89	0.69	0.70	27.14	1.12	0.85	0.89	70.89
D.4 TidalTrust	0.96	0.73	0.74	27.86	1.08	0.82	0.83	74.67
E. Our recommender								
E.1 Biased RW-RS	0.78	0.61	0.59	92.61	1.07	0.82	0.83	53.43

Table 3 Predictive accuracy metrics: results on cold-start users

Datasets	Filmtrust				Epinions			
Performance metrics	RMSE	MAE	MAUE	Coverage (%)	RMSE	MAE	MAUE	Coverage (%)
A. Baseline								
A.1 Random	1.51	1.22	1.22	100.00	2.00	1.67	1.67	100.00
A.2 Always-Max	0.80	0.49	0.51	100.00	1.56	0.94	0.93	100.00
B. Collaborative filtering								
(All neighbors)	0.80	0.64	0.63	82.98	1.09	0.82	0.82	69.46
C. Content-based recommendation								
(Nearest n items)	0.77	0.63	0.64	72.60	1.58	1.09	1.08	9.21
D. Trust-based approaches								
D.1 MoleTrust-1	1.46	1.20	1.02	10.94	1.49	1.09	1.09	7.49
D.2 MoleTrust-2	1.71	1.33	1.08	20.41	1.53	1.17	1.17	24.27
D.3 MoleTrust-3	1.22	0.87	1.33	24.56	1.06	0.82	1.08	76.25
D.4 TidalTrust	1.22	0.87	0.87	26.23	1.11	0.84	0.84	42.00
E. Our recommender								
E.1 Biased RW-RS	0.83	0.62	0.61	76.92	1.1	0.85	0.86	40.29

$$MAUE = \frac{\frac{1}{|M|}\sum_{u=1}^{|M|} |\widehat{r_{u,i}} - r_{u,i}|}{N} \tag{7}$$

where M are each distinct user's rating and N their overall number in T.

The *ratings' coverage* measures the percentage of ratings in the test set for which the system manages to make a prediction. It should be pointed out that an RS that exhibits satisfactory results in the statistical accuracy metrics is still considered to perform poorly if it manages to produce recommendations only for a handful of users or items. More formally, the *rating's coverage* is defined as

$$\text{Coverage} = 100\frac{|T_R|}{|T|} \tag{8}$$

where $|T_R|$ is the cardinality of the set of the items for which the RS produced recommendations (generally, $T_R \subseteq T$).

Finally, it should be pointed out that the performance of the RS on the predictive accuracy metrics has been tested in two different user sets. The first one (Table 2) naturally includes all the users in the dataset while the second (Table 3) only includes the *cold-start users*; those users who have provided five ratings or less [15]. Cold-start users constitute a large portion of the user base of a RS and they could be viewed as the "newcomers" in the system. Definitely, the RS should be able to propose meaningful items from the very beginning in order to gain their confidence. Lastly, for the results presented in Tables 2 and 3, the standard deviation for RMSE, MAE, and MAUE was in the range of 0.01–0.03 for all systems, datasets, and views and in the 1–2 % range for coverage.

6.2 Classification Accuracy

Classification accuracy metrics estimate the quality of the recommendations by measuring how frequently the RS makes good predictions [11]. This category of metrics is not evaluated on single withheld ratings but rather on a list of recommended items; for this reason the experimentation protocol has to be modified. Instead of splitting the whole dataset into disjoint sets, only the ratings of a particular target user are extracted and split into a training and a test set of a specific size (5 through 25 items in our experiments). Then, an equally sized list of items is presented to the target user and evaluated by the protocol. This process is repeated iteratively for all users and the results are averaged over all runs (Figs. 2 and 3). Since the recommendation list has to be of at least a certain size in order for the computations to be legitimate, this protocol cannot be run on the cold-start users alone. The results of the baseline systems are also not displayed because they have exhibited almost zero performance on this set of metrics.

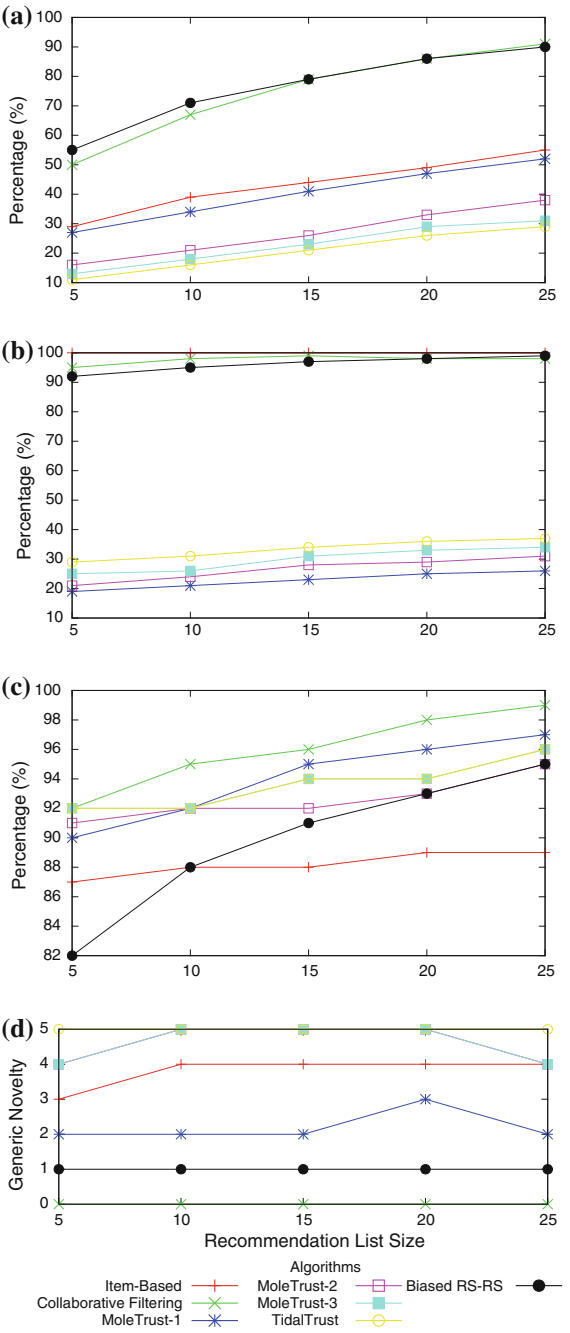

Fig. 2 Classification accuracy metrics on the Filmtrust dataset. **a** Precision. **b** Reach. **c** Intra-list diversity. **d** Popularity-based item novelty

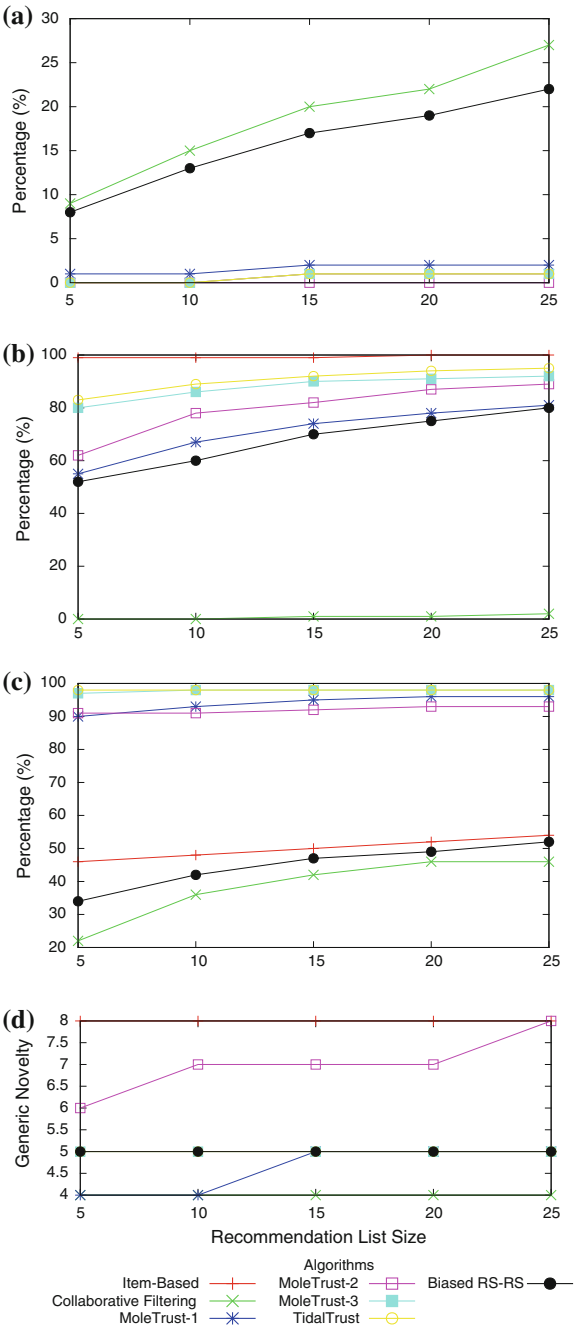

Fig. 3 Classification accuracy metrics on the Epinions dataset. **a** Precision. **b** Reach. **c** Intra-list diversity. **d** Popularity-based item novelty

The most common metric in this category is *Precision*. It measures the proportion of the relevant items selected in the list (N_{rs}) versus the total number of the selected items (N_s)

$$\text{Precision} = 100\frac{N_{\text{rs}}}{N_s} \tag{9}$$

Alternatively, precision may be viewed as the probability that a selected item is relevant and it is most commonly expressed as a percentage.

A metric similar to the ratings' coverage discussed in Sect. 6.1 is *Reach*, or the percentage of the users for whom the RS manages to produce recommendations. Again, a recommender system that exhibits a high precision in its proposals is still considered to perform poorly if it manages to do so only for a handful of users. More formally, *Reach* is defined as

$$\text{Reach} = 100\frac{|U_R|}{|U|} \tag{10}$$

where $|U_R|$ is the cardinality of the set of the users for which the RS produced recommendations and $|U|$ is the cardinality of the set of users in the system (generally, $U_R \subseteq U$).

6.3 Novelty and Diversity

This set of metrics tries to quantify the obviousness and the ordinarity of the recommendations a particular user receives. Generally, commonplace recommendations are considered to be of low quality even if they are correct in terms of both the prediction and the classification accuracy [11]. These two metrics are evaluated on a list of recommended items and for this reason the experimentation protocol of the previous subsection is applied and the results are displayed on the same set of Figs. 2 and 3

Novelty measures the extent to which an item (or a set of items) is new when compared with those items that have already been consumed by a user (or a community of users). Several models of item novelty have been proposed in the literature; in our experiments, we have used *the generic popularity-based item novelty* [16], which is defined as

$$\text{novelty(i)} = I(i) = -\log_2 p(i) \tag{11}$$

where $p(i)$ is the probability of observing item $i \in I$ in the result set. In our case, we considered this probability to be analogous to the number of ratings this item has received ($|R_i|$) proportional to the total number of ratings in the dataset ($|R|$)

$$p(i) \sim \frac{|R_i|}{|R|} \tag{12}$$

Diversity, on the other hand, measures how different the items of a recommendation list are from one another. A list of items that are relevant but very similar to each other is considered to be very ordinary and thus of low quality. In our experiments, we have used the *Intra-list Diversity* metric defined as

$$\text{diversity}(L) = \frac{2}{|L|(|L|-1)} \sum_{k<n} d(i_n, i_k) \tag{13}$$

where L is the list of the recommended items, $i_n, i_k \in L$ and $d(i, j)$ is an item distance measure. As we have been using the Manhattan distance measure which takes values in the [0, 1] interval, the results of the intra-list diversity are normalized on the percentage scale (Figs. 2 and 3).

7 Results

A first observation is that the *Biased RW-RS* algorithm is comparable to the collaborative filtering approach in all of the predictive accuracy metrics on the whole users view, despite being a social method. In general, Social RS exhibit poor behavior on coverage and this is attributed to the fact that the trust network in both datasets is very sparse; as a result, their exploration ability is greatly impacted. However, the *Biased RW-RS* manages to overcome this difficulty by probabilistically deciding at each step to either pick a trust neighbor or a similar user. Therefore, it is far superior in terms of coverage in the Filmtrust dataset and on the average of the SRS in the Epinions dataset.

Furthermore, for the cold-start users, the *Biased RW-RS* algorithm is among the most efficient approaches in the Filmtrust dataset for this special case of users, clearly outperforming the other social methods, while the performance of the other RS (traditional and social) deteriorates evidently. Again, in the Epinions dataset, our algorithm manages to keep a steady performance in terms of the MAE and RMSE metrics, while at the same time offering an adequate coverage on the ratings of the test set.

The *Biased RW-RS* algorithm is the most efficient social method at the precision metric on both datasets. Although collaborative filtering seems to be slightly better in the Epinions dataset, it performs very poorly on the reach metric (around 1–2 % on all list sizes), meaning that it is able to produce accurate recommendations only for a tiny slice of the users. Trust approaches, on the other hand, are able to produce recommendations for more users; however, these predictions are far from accurate (precision is less than 5 % for all trust metrics on all list sizes on the Epinions dataset and less than 40 % on the Filmtrust dataset) because user correlation is not taken into account. Another notable observation for the trust approaches is that their Reach on the Filmtrust dataset is about one-third compared to the Epinions dataset, even though the trust network of the former dataset is denser than the latter (global clustering coefficient characteristic of Table 1). This phenomenon is attributed to

the fact that only 38 % of the users in the Filmtrust dataset participate in the trust network, while the same figure for the Epinions dataset is 68 %. As a conclusion, even the smallest user engagement in the trust network is sufficient for the SRS to make recommendations.

The trust approaches also demonstrate the best results in terms of both the novelty of the recommendations and the diversity of the items in the recommendation list. However, this behavior should not be studied independently from Precision; diverse and novel predictions are of no use if they are not relevant to the user. On the other hand, correlation-based approaches (item-based and collaborative filtering RS) make recommendations of items that are very obvious and to a large extent very similar to one another. For this reason, these systems exhibit very poor novelty, which is also illustrated in the respective Figs. 2d and 3d.

In all, the results indicate that our system achieves recommendation accuracy similar to the traditional collaborative approaches while showing better novelty and diversity, due to the incorporation of social aspects in its recommendation mechanism.

8 Conclusion

In this chapter, we have presented a novel approach toward SRS, a random walk recommender system based on rejection sampling. Our contribution is an algorithm, *Biased RW-RS*, which is based on a novel idea in neighborhood selection; it deviates from the standard view of all trust neighbors as equally probable and models their similarity to the target user as a probability distribution. Since this probability distribution is unknown for each user, it is approximated by using readily applied tools from the statistical literature and more specifically of the rejection sampling algorithm. Generally, the results on the reference datasets are encouraging and in accordance to our claims.

We are also taking into consideration the fact that our system does not exhibit a steady performance lead in the Epinions dataset. We attribute this behavior to the peculiarities of this specific dataset; its greater sparsity and the fact that it is not domain specific (as opposed to the Filmtrust dataset). Therefore, our algorithm should be further adapted in the direction of addressing the aforementioned observation.

References

1. Jamali M, Ester M (2009) TrustWalker: a random walk model for combining trust-based and item-based recommendation. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. KDD'09, New York, NY, USA, ACM, pp 397–406
2. Abbassi Z, Mirrokni VS (2007) A recommender system based on local random walks and spectral methods. In: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop

on Web mining and social network analysis. WebKDD/SNA-KDD'07, New York, NY, USA, ACM, pp 102–108
3. Yuan Q, Chen L, Zhao S (2012) Augmenting collaborative recommenders by fusing social relationships: membership and friendship. In: Recommender systems for the social web, vol 32 of Intelligent Systems Reference Library, Springer, Berlin, pp 159–175
4. Zhang Y, Wu Jq, Zhuang Yt (2009) Random walk models for top-n recommendation task. J Zhejiang Univ Sci A 10:927–936
5. Singh AP, Gunawardana A, Meek C, Surendran AC (2007) Recommendations using absorbing random walks. In: North East Student Colloquium on Artificial Intelligence (NESCAI)
6. Golbeck JA (2005) Computing and applying trust in web-based social networks. Ph.D. thesis, College Park, MD, USA, AAI3178583
7. Andersen R, Borgs C, Chayes J, Feige U, Flaxman A, Kalai A, Mirrokni V, Tennenholtz M (2008) Trust-based recommendation systems: an axiomatic approach. In: Proceedings of the 17th international conference on world wide web. WWW'08, New York, NY, USA, ACM, pp 199–208
8. Konstas I, Stathopoulos V, Jose JM (2009) On social networks and collaborative recommendation. In: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval. SIGIR'09, New York, NY, USA, ACM, pp 195–202
9. Alexandridis G, Siolas G, Stafylopatis A (2013) A biased random walk recommender based on rejection sampling. In: Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM 2013), Niagara Falls, Canada
10. Massa P, Avesani P (2007) Trust metrics in recommender systems. Int J Semant Web Inf Syst
11. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22(1):53
12. Golbeck J, Hendler J (2006) FilmTrust: movie recommendations using trust in web-based social networks. In: Consumer communications and networking conference, 2006. CCNC 2006, 3rd IEEE, vol 1, pp 282–286
13. Massa P, Bhattacharjee B (2004) Using trust in recommender systems: an experimental analysis. In: Proceedings of iTrust2004 international conference, pp 221–235
14. Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithm for collaborative filtering. In: Proceedings of the 14th conference on uncertainty in artificial intelligence, pp 43–52
15. Massa P, Avesani P (2009) Trust metrics in recommender systems. In: Golbeck J (ed) Computing with social trust, Human Computer interaction series. Springer, London, pp 259–285
16. Castells P, Vargas S, Wang J (April 2011) Novelty and diversity metrics for recommender systems: choice, discovery and relevance. In: International workshop on diversity in document retrieval (DDR 2011) at the 33rd European conference on information retrieval (ECIR 2011)

Social Network Derived Credibility

Erica J. Briscoe, Darren Scott Appling and Heather Hayes

Abstract The increasing use of social media results in users that must ascertain the truthfulness of information that they encounter from unknown sources using a variety of indicators (e.g., explicit ratings, profile information, etc.). Through human-subject experimentation with an online social network-style platform, we focus on the determination of credibility in ego-centric networks, where participants are able to observe salient social network properties, such as degree centrality and geodesic distance. Using manipulated social network graphs, we find that corroboration and degree centrality are most utilized by subjects as indicators of credibility. utilized by subjects as indicators of credibility. We discuss the implications of the use of social network structural properties, use principal components analysis to visualize the reduced dimensional feature space, and analyze how credibility changes per property according to the "Big 5" theory of personality.

Keywords Social networks · Structural properties · Credibility · Trust · Human experimentation

1 Introduction

Many approaches for investigating social media behavior stem from sociology, e.g., [1], graph theory, e.g., [2], or some hybrid combination, e.g., [3]. Our work follows a hybrid methodology by utilizing an experimental approach to determine how users of social media estimate credibility in light of explicit structural information that they have about their online social network graphs. We intentionally investigate the effects of structural properties in isolation from other forms of social network

E.J. Briscoe (✉) · D.S. Appling · H. Hayes
Georgia Tech, 220 North Avenue, Atlanta, GA, USA
e-mail: erica.briscoe@gtri.gatech.edu

D.S. Appling
e-mail: scott.appling@gtri.gatech.edu

H. Hayes
e-mail: heather.hayes@gtri.gatech.edu

Ö. Ulusoy et al. (eds.), *Recommendation and Search in Social Networks*,
Lecture Notes in Social Networks, DOI 10.1007/978-3-319-14379-8_4

information, such as explicit credibility assignments, e.g., "badges," and "likes" [4], user profiles, e.g., [5], network dynamics, e.g., [6], and sociolinguistic properties, e.g., [7]. The motive for this concentration stems from our desire to evaluate how people piece together the available information on social networking sites, so as to avoid overfitting classification algorithms without regard to users' psychological motivations for feature selection; however, this work is intended to inform (and improve the generalization of) such classification algorithms.

Our work differs from other investigations into trust and trust propagation (e.g., [8–10]), as we explore individuals' judgments of credibility as they make their determination through the observation of ego-centric network properties in isolation from other traditional indicators. Here, credibility is not evaluated over long-established friend networks, whose history might contain additional information. We are interested, rather, in the isolated study of structural social network properties that people utilize in the absence of "real-world" relationship knowledge to determine the credibility of information from their alters. This approach is supported by the evidence of a differentiation between *cultural* and *structural* schemata, which are suggested to guide the encoding of social network information [11]. In popular social media, a concentration on structural features is likely to appear when relationships are new or superficial in nature, such as when users seek to acquire many more "friends" than they actually know or possess in the offline world. The existence of a potential cognitive limit to the number of people with whom one can maintain stable social relationships [12] may also suggest the unlikely application of the same mechanisms that people use for long-lasting and "deeper" relationships to many of those that exist in social media.

Here, a definition of terms is necessary, as many common terms are often used interchangeably. In many studies, *trust* is the relying party's subjective belief in an entity to serve a certain function, such as information provision [8]. *Reputation*, in this context, implies permanence of a relationship and is therefore not relevant to our study that concentrates on immediately perceivable social network properties. We identify the property that we are interested in here as *credibility*. While similar to trust, we feel that credibility implies a determination of the truthfulness of content as originating from a person, as opposed to the general trustworthiness of that person.

2 Utilizing Social Information

Experiments involving trust have been carried out in a number of contexts where trust is important to the basic function of the particular platform, for example, the evaluation of sellers' trustworthiness on e-commerce sites such as eBay [3]. Here, trust is garnered by using parties' feedback as input into a reputation system that provides a globally visible score. Other algorithmic approaches utilize properties of networks to determine trust, such as PageRank [13], where reputation is derived from the number of hyperlinks to a Web page. Aggregation methods may calculate combined trust values over various weighted links [9], where [14] finds that the

shortest and strongest trust paths are the best predictors of the level of trust. The most direct measures of trust involve asking users to explicitly rate users in their network in terms of their trustworthiness [15].

In contrast, we are especially motivated by how people estimate credibility in situations where information originates from previously unknown contacts, which do not merely mirror real-life relationships. These occasions are also likely to arise in times of specific, significant events, such as in case of humanitarian disasters, where information is provided by those physically close to an event, rather than non-virtual close friends. These types of superficial friendships are also often seen on platforms that allow for the visible dissemination of information and that explicitly identify relationships, such as on Twitter. For example, a Twitter user may receive a "tweet" from someone they started following because someone they knew was following that person.

Previous studies [16, 17] have also included structural factors in analyses of user behavior. As some studies, and perhaps popular belief, suggest, we expect that the amount of friends that a person is known to have (often referred to as their popularity) is highly relevant to determining their credibility (similar to that found in studies of influence in social networks (e.g., [18])).

To better understand exactly what features people attend to and utilize when making credibility decisions, we perform human-subject experimentation in which a simulated social media platform is used to allow subjects to interact and indicate their judgments based on visible interactions. While recognizing that people utilize a combination of information in making decisions about credibility, rather than analyzing these decisions in light of all available information, we provide subjects with only structural properties to utilize in a task in which they must determine other users' credibility. The properties on which we concentrate have been gleaned from previous studies concerning the effects of structural variables (e.g., [17]). It should also be acknowledged that in this particular study, we do not focus on the potential for deception, rather, their choices are focused on the question 'who do you most believe', as opposed to 'who do you think is being untruthful'.

3 Method

3.1 Participants

Forty-nine participants were recruited from the Georgia Tech student population and paid for their participation in the study.

3.2 Materials and Procedures

The experiment described here was a task in a larger study of social media usage in which participants were asked to use a mock social media site, "FaceFriend," to observe and communicate with other users. The mock site was intended to resemble a popular social networking site (see Fig. 1). Participants completed three pre-experiment surveys, including a personality test and a survey regarding their social media and email usage (another survey not relevant to these results was also conducted). We also asked users about their email usage to provide a contrast for social media and to observe differences in attitude that may indicate dissimilar behaviors for this communication medium.

Images displaying online social network graphs, characterized as undirected graphs where nodes represented one ego and its alters, were created for use in the study. A tie between any two nodes indicated a friendship relationship. These networks were constructed such that a specific structural property for each commenting alter (only alters commented) (e.g., their degree centrality) varied from other alters in the network. For each of these alters, the value of the manipulated property (for example, degree centrality) was varied in the direction that had been determined

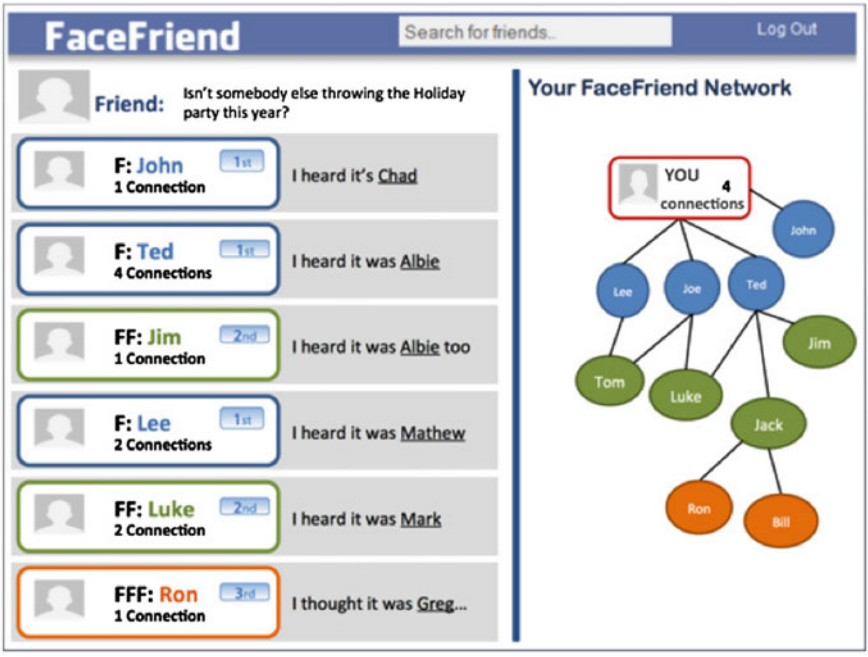

Fig. 1 A social media "wall" that was presented to subjects depicting a conversation thread and the undirected social network graph that described the relationships among the people commenting in the thread

Table 1 Structural properties intentionally varied across commenters

Node property	Operational description
Geodesic distance *,**	The number of edges in the shortest path connecting two vertices (between the node representing the subject and the commenter).
Degree centrality *,**,***	The number of edges incident upon a node.
Consensus **,***	Agreement by two commenters in a sub-graph (relative to the subject).
Substantiation ***	Agreement in the network by two nodes in separate sub-graphs (relative to the subject).
Overlap **	The number of paths to a commenter from the subjects node.

*, **, *** Superscripts indicate trials in which quantities were manipulated

most expressive of credibility (through pilot studies). Table 1 describes each of the properties. While these are not the only structural properties exhibited within the network, these are the properties suggested by past studies, e.g., [14], and our own pilot studies as perceived as credibility indicators by viewers. To determine the effect of co-occurring high-credibility qualities, in Trial 2, a commenter's node was manipulated so as to present high values across two structural properties (degree centrality and geodesic distance).

Before the experiment, each subject was provided with a depiction of a social network (similar to that on the right in Fig. 1) and was told how, for this study, it represented the set of his or her relationships on the social networking site. The relationships were briefly described to the subjects (e.g., they were told that a node connected to their node was a "friend"). Network properties, such as the degree centrality or multiplex relationships (e.g., consensus between commenting alters), were not explicitly identified to the subjects, as we are interested in how individuals naturally utilize these properties without being directed. Relationship weights, or tie-strengths, of edges were also not indicated, so that subjects had no reason to believe that any one relationship was stronger than any other.

Subjects were then asked to view a series of conversation "threads" (see Fig. 1) on a computer monitor. The threads were explained to be the same as any normal conversation on social media, such as that as would be observed on a person's "wall". Each thread was presented along with an accompanying social network that depicted the relationships of the people who commented in the thread. Each commenter in the thread presented his or her opinion regarding the question posed in the initial comment. For example, an initial question was "Who got the new car I saw in the parking lot?" Each question was followed by various responses by the commenters such as "I heard it was Josh". The language used in the comments was controlled so as to avoid sociolinguistic variance, which has been shown to affect credibility determinations [7]. Profile images were not included so as to prevent bias on the basis of an image; likewise, names were randomly chosen from the top 50 currently most popular American names (as published by the US Social Security Administration), to avoid cultural bias. To evaluate for potential spatial biases, each trial was replicated using a different spatial configuration.

After every trial, each with its own thread and social network graph, the subjects were presented with the list of the names that were suggested by the commenters in answer to the question that started the topic thread. They were instructed to order those names in terms of who they thought was the correct answer to the question that was posed in the thread. For example, for the thread presented in Fig. 1, the subject might respond with "Albie, Mathew, Mark, Greg, Chad" to indicate who he believed was throwing the holiday party this year. It is important to note here that the names provided as possible answers to the posed question were not contained within the social network. The subjects were asked to choose the names based on the persons who suggested them, where the only information that they had about these persons was the social network in which both the subject and the commenters were situated. After making their selection, subjects were asked to enter text explaining their ranking. For example, after choosing "Chad" as the most credible (indicated by placing this name first in a ranked list) a participant might enter "I chose Chad first because John said that it was him and John is a close friend of mine."

4 Results

Results from the social media usage survey indicated that all subjects regularly used social media. Forty-nine subjects saw Trials 1 and 2. To ensure that the intended network properties were perceived and used in the choices, we also qualitatively analyzed the free-form responses that the subjects provided. Trial 3, added during the experimentation based on free-form responses that indicated that subjects perceived corroborating evidence differently depending on if it came from unrelated sources, had a total of 38 subjects. Free-form responses indicated that the subjects did indeed perceive the targeted properties. Table 2 presents examples of free-form responses that informally describe the targeted properties that were used by the subjects to make their credibility decisions.

Table 2 Examples taken from subjects' free-form responses that reflect each of the targeted network properties

Property	Referring statement
Geodesic distance	"I am closer to Kate so I may trust what she says more"; "Because I believe that my close friends to more truthful"
Degree centrality	"...Ray does not know too many people."
Consensus	"Multiple member agreement"; "...mentioned twice with friends from different degrees"
Substantiation	"two unrelated people [sic] its Sam so it increases the probability"; "Two independent sources."
Overlap	"..is connected to two of my level one friends."; "he is a friend of a friend that is connected to my two other friends"

Table 3 Values of social network properties for named nodes in each trial

Trial	Node name	Geodesic distance	Degree centrality	Consensus	Substantiation	Overlap
1	Lee	1	5	0	0	1
1	Jack	2	3	0	0	3
1	Joe	1	2	0	0	1
1	Jim	2	2	0	0	1
2	Bob/Hal	4	4	2	0	2
2	Mel	1	2	0	0	1
2	Sam	1	1	0	0	1
2	Ray	2	2	0	0	2
2	Dan	3	0	0	0	1
3	Carl/David	3	4	2	2	1
3	Tony / Frank	3	2	2	0	2
3	Pat	2	5	0	0	1

Table 3 is a listing of the network property values that were set for each trial. For example, a 2 in the substantiation and consensus column indicates that there were two nodes that provided the identical information, either in separate sub-graphs or the same, respectively.

To determine preferences in terms of network properties that signal credibility, we analyzed the rankings that the subjects provided using a Friedman's Test to determine if the participants demonstrated a differential rank-ordered preference for particular nodes based on the manipulated node qualities. We associated a number rank to correspond to each node's place in the ranking (based on subjects' choices on the answer to the displayed questions mapped to those nodes who provided the answers), creating an ordinal-dependent variable. The mean ranks resulting from the analysis are displayed in Table 4. These ranks indicate the preferences of the subjects in terms of which nodes they felt were most credible, where a "1" indicated the most credible rank (when the name was listed first by the subject in the trial).

The Friedman test indicated significant differences in the rankings. For trials 1 and 2, $N = 49$, $\chi^2(3) = 38.265$, $p < 0.0001$ and $N = 49$, $\chi^2(4) = 68.882$, $p < 0.0001$, respectively. For trial 3, $N = 38$, $\chi^2(3) = 38.895$, $p < 0.0001$. Because the test does not indicate where the differences occur, we ran post hoc pairwise comparisons. All paired ranks were significantly different (using a post hoc Wilcoxon Rank test ($p < 0.001$), for which familywise error was controlled), except for between Overlap and Geodesic distance in Trial 1 and 2 ($p = 0.228$ and $p = 0.102$, respectively). All pairwise comparisons were significant in Trial 3. See Table 4 for the mean ranks of each targeted (manipulated) node across trials.

Table 4 Mean ranks of targeted network properties

Trial	Manipulated property of ranked node	Mean rank
1	Degree centrality	1.67
1	Overlap	2.37
1	Geodesic distance	2.71
1	Contrast	3.24
2	Consensus	1.67
2	Centrality & Geodesic distance	2.55
2	Geodesic distance	3.16
2	Overlap	3.47
2	Contrast	4.14
3	Consensus	1.39
3	Substantiation	1.82
3	Degree centrality	2.79

5 Rankings and Structural Properties

The intent of this work is to demonstrate the explicit use of social network information by people who are forced to make a credibility decision with no contextual or historical relationship information available. Our results show that without additional social context, people somewhat consistently estimate credibility based on network properties (such as the geodesic distance from an individual to the person who provides him information), where *consensus*, or having more than one person in the same sub-graph agree on a claim, is the predominant indicator of credibility. As hypothesized, high degree centrality of a node is also a highly-utilized feature toward determining credibility, where, for example, a subject felt that "Pat has the highest number of connections, so she must be right." Qualitative analysis of the free-form responses confirmed the use of the targeted qualities. For example, one comment was I believe my closer friends are more truthful, which is indicative of recognition and usage of the geodesic distance property.

Trial 3 was specifically designed to test how people weighted corroborating evidence faced with that evidence either coming from the same sub-graphs, which could be explained by the piece of corroborating information resulting from the propagation of that information, or from separate sub-graphs, where two independent sources agree. Recognition of this distinction was also observed in the free-form responses (e.g., one response was "David probably heard from Carl"). Here, *consensus*, or corroboration within a sub-graph, was overall more highly ranked. Free-form responses also show additional combinatorial effect of structural properties (e.g., "I chose the person with the most amount of connections and how close they were to me and how many people gave the same answer"). This combination was also evident in the high ranking of the combined high-value node in Trial 2, indicated as "Centrality & Distance" in Table 4.

To better understand how subjects may be combining across social network features, we coded each presented node in each trial network according to its property values. Here, we coded substantiation as a binary valued feature indicating whether a neighbor node suggests the same answer to the posed question. Node overlap was coded as the number of separate paths from the subjects node to the presented node. Consensus is the number of nodes in the presented node sub-graph who share the same answer in the thread. Distance was coded as the number of edges from the ego to the commenting alter and centrality was simply the degree centrality (see Table 3).

Next, we visualized the relationship between these features and the rank information collected from participants. We performed principal components analysis (PCA) on the presented node set given our node representation (degree, substantiation, overlap, consensus, and distance). We retained the top three principal components based on their eigenvalues and those accounting for over 95 % of the variance. We projected the points (presented nodes) through the reduced eigenvector matrix to produce a reduced feature representation. For each point (node) we use the normalized sample mean rank to indicate the rank of a node. Figure 2 shows the scatterplot of the points in the reduced space, where the normalized mean rank of nodes is indicated by color: blue indicates highly ranked, pink lower ranked items. Here, we see similar node ranks clustering in the reduced space based on the three leading principal components.

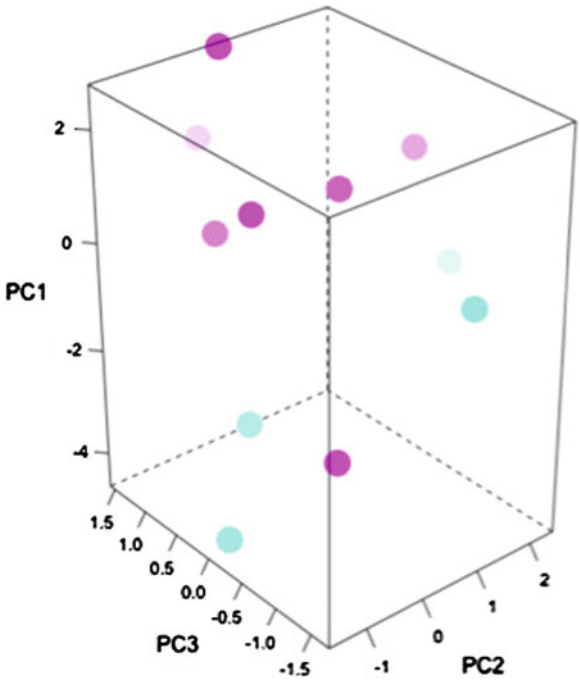

Fig. 2 Principal Component Analysis using social property values of nodes in a three-dimensional space with *rich blues* indicating higher rankings and *deep purples* indicating very low ranked items

The components with their coefficients (how much weight each contributes) are as follows: the first component primarily reflects degree centrality (0.93) and consensus (0.28). The second component reflects geodesic distance (0.72) and consensus (0.55). The third component is composed of overlap (0.92), consensus (0.31), and substantiation (0.21). Highly ranked points tend to have a negative association with the first component. Lowly ranked nodes tend to have high values on the first component and low values on the third component.

6 Credibility Ranking Discussion

The intent of this work is to demonstrate the use of structural social network information by people who must make a credibility decision with no contextual or historical relationship information available. This investigation follows from work that suggests that information about social networks is perceived and potentially processed differently from other types of social information [19]. Our statistical results show that, without additional context, people estimate credibility based on network properties (such as the geodesic distance from an individual to the person who provides him information) in a principled manner, where consensus—having more than one person corroborate a claim where the corroborators are in the same sub-graph relative to the decision maker—was significantly chosen as the highest indicator of credibility (highest ranked in both trial 2 and 3). When two high-valued qualities co-occur, such as Degree Centrality and Geodesic Distance in Trial 2, it suggests that subjects combine the "value" of multiple structural properties. This effect was confirmed, qualitatively, by the subject free-form responses. For example, the comment "I know a person who knows Ray and Ray knows lots of people" is indicative of the recognition of two structural qualities.

While realistic evaluations of credibility may often not involve explicit reference to a social network graph, these results may be evaluated in light of work that investigates how people utilize different schemata or mental frameworks that allow for the organization and processing of information involved in memory formation and recall [11]. When people are required to make decisions about information that they receive from people in their network, they utilize structural schemata to recover network properties that they had encoded, based either on explicit social network information or through other equivalent indicators, such as follower counts in Twitter. Our experimental results and dimensionality reduction suggest how these recalled properties may be applied and potentially utilized in a credibility prediction algorithm. A caveat, however, is the use of the term "credibility" as operationalized here, does not necessarily conform to other definitions or applications (see [20] for further discussion), but rather is specifically focused on determining the potential truthfulness of information (without regard to deception) as determined by observing a social network.

7 Testing Degree Centrality Magnitude

In our previously discussed experiments, specific levels of centrality, based on network sizes ranging from 1 to 5 connections, were used. Our rationale for this variance was that, within the laboratory-induced time constraint, this range would be tractable for subjects to reason about along with the other network properties. To rule out the possibility that the mean rankings observed in Trials 2 and 3 were conditioned upon these specific amounts of connections used for the centrality conditions, we conducted a follow-up study to determine the effect of relatively large magnitudes of variance in the degree centrality of nodes. While consensus was found to be the most indicative of credibility in Trials 1–3, we expect that if degree centrality was allowed to increase to extreme magnitudes, that there would be a point of cross-over, where the subjects would start choosing the nodes with extremely high degree centrality. To investigate the presence of such an effect, we based degree centrality on more realistic distribution, typical among current social network users. Using a power-law to model the distribution of the number of connections, with a minimum of 0 and a maximum of 300,000, we made 12 uniform cuts based on the incremental distribution of the probability mass accounted for at each cut point (8.3 %). These values were then used as the number of connections (i.e., "friends") that nodes had in new experimental trials, wherein participants were presented with the decision to choose as more credible a commenter who demonstrated degree centrality as determined by cut points or a commenter who demonstrated the consensus property (i.e., the commenter's information was agreed upon by another node in the sub-graph). The consensus-representative node had a controlled centrality always less than the other node. Otherwise the procedure was the same as described in Trials 1–3 above.

A one-way ANOVA was conducted to ascertain whether there was an effect of degree centrality on the choice of the most credible node. The resulting omnibus test statistic ($N = 20$, $F = 0.303$, $p > 0.05$) was below the critical value and the null hypothesis was retained; we infer that the different magnitudes of centrality values held no effect on determining whether or not consensus was chosen as the more credible network property, i.e., the consensus feature trumped the centrality degree-focused feature at all degree levels.

8 Effects of Personality on Credibility Assessment

While our previous series of experiments focused on the assessment of credibility by individuals in general, we now focus on the degree to which psychologically motivated, trait-based theories can be used as an indicator of credibility decisions. We are interested in how the ranking of nodes in social networks, in terms of their credibility and as characterized by specifically manipulated node qualities, will be predicted by the decision-maker's personality traits. In the next section, we succinctly

review the literature on personality characterization and describe our hypotheses on the use of personality dimensions toward predicting credibility rankings in a social network.

8.1 Personality

The effect that individual characteristics play in determining a person's cognitive processes and behaviors is often investigated through studies involving the characterization of personality traits. These traits are generally assumed to be temporally stable and cross-situational. The most frequently used psychological approach for studying personality is the five-factor model [21], often referred to as the Big Five [22] dimensions of personality. These factors are dimensions, not types, so every person's personality can be identified as existing somewhere on each of five continua. These factors are considered cross-cultural [23] and knowing a person's placement on each dimension may serve useful for predicting behavior and tailoring approaches (e.g., [24]). Below, we briefly describe the five personality dimensions and examples of related research exploring their relationship to computer-mediated communication.

8.1.1 Openness

This dimension reflects a person's inclination to be curious, insightful, and have wide interests [25]. Those who measure high on the Openness dimension are likely to be receptive to new ideas and to consider multiple perspectives; those who score low avoid consideration of ambiguous content. Barrick and Mount [26] found that social media usage is related to high openness.

8.1.2 Conscientiousness

This factor contains aspects of dependability and an achievement orientation [27]. Usually highly conscientious people are known to be hardworking, ambitious, and organized; low scorers on this dimension are impulsive.

8.1.3 Extraversion

Perhaps one the most well-known dimensions, this attribute represents an individual's propensity to be talkative, outgoing, sociable, and energetic [25]. These people are likely to more dominant compared to introverts, who are often quiet, reserved, and withdrawn. Correa et al. [28] found that extraversion is positively related to social media use, perhaps related to the fact that computer-mediated communication might cause some to be more outgoing, especially those who are introverted [29].

8.1.4 Agreeableness

This factor reflects the friendliness and cooperativeness of individuals. High scores seek social harmony, where low scores are suspicious of others. A study by [30] examining Facebook patterns of usage found that openness was positively correlated with the number of users likes, group associations, and status updates.

8.1.5 Neuroticism

This factor also referred to as emotional stability, captures anxiety, anger, depression, and instability. Those low on the scale exhibit confidence, calmness, and poise [25]. Previous work has found that those scoring high on the neuroticism dimension use social media more than those who score lower [28].

8.2 Personality and Credibility Assessment

Using the experimental design previously described (utilizing the 'Big 5' personality survey that was administered to all subjects), we conducted human subjects experimentation 72 participants from Georgia Tech; ages ranging from 18 to 47 years, mean 23 years (SD = 5.7); 53 % female. We pair participants observed personality scores with their credibility ranking decisions to empirically test our three hypotheses regarding the effect of personality on determining credibility using social network properties, as described below.

H1: Those scoring high on openness will rank nodes (commenters) that exhibit higher degree centrality more credible, reflecting their attention to multiple perspectives.

H2: Because of their greater tendency toward sociability, subjects will higher levels of extraversion will rank nodes with increasing levels of degree centrality and network size, higher.

H3: Because of their greater tendency toward harmony and cooperation, subjects with greater levels of agreeableness will rank nodes with lower (closer) geodesic distance, as being more credible.

For the remaining two dimensions, it is not clear in this context how they will result in an effect on the subjects in terms of their preference of any node property over another for credibility determinations.

To determine dominant preferences in terms of network properties that signal credibility, we conducted 5 series of ANOVAs (one for each property type). Within each series we conducted 5 ANOVAs, one for each of the personality dimensions, including age and gender. The dependent variable for each of the 5 series was the sum of the highest ranked node's particular network property value for all trials (e.g., the sum of the degree centrality values across all nodes ranked first would constitute the

Table 5 Main effect sizes for network properties by personality dimension

Dimension	Network size	Geodesic distance	Overlap	Consensus	Centrality
Openness	**0.193***	0.042	**0.160***	0.096	0.096
Conscientiousness	0.033	0.002	0.045	0.019	0.019
Extraversion	0.021	0.035	0.022	0.035	0.035
Agreeableness	0.031	**0.143***	0.132	0.126	0.126
Neuroticism	0.027	0.098	0.029	0.045	0.045

* indicates significant effect at $p < .05$

combined centrality-dependent measure; likewise with the other properties). The age and personality dimensions were binned into 4 levels using $\pm$ 1 standard deviation to set the cut points.

The main effect of the openness dimension was statistically significant according to the Omnibus F-test ($p < .001$). To further investigate H1, we followed up with a Tukey test and found that individuals with higher openness scores tended to choose nodes with higher centrality property values ($p < .05$). A main effect was also found for geodesic distance and the agreeableness dimension ($p < .038$). A Tukey test revealed that the mean credibility rankings were lower for individuals with lower agreeableness scores ($p < .001$), consistent with the second claim in our third hypothesis. The full set of results is displayed Table 5.

These results show evidence for a subset of the specific effects that we evaluated. We found the existence of a relationship between extraversion and centrality, agreeableness and both centrality and geodesic distance, and openness and centrality. This result complements a line of work that suggests that information about social networks is perceived and potentially processed differently from other types of social information [31]; differences that may be further explained through individual characterization, i.e., personality.

We see the greatest value in this investigation stemming from its potential application within more robust algorithmic approaches, such as in "fact-finder"-type algorithms (e.g., [32]) and trust prediction approaches (e.g., [33]). Here, the incorporation of parameters, such as those representing network properties (e.g., degree centrality) that may be discovered through intermediary models of social media users, offers the potential for significant performance improvement. This is especially promising given the success of methods for automatically determining personality dimensions through the analysis of user-produced content (e.g., [34]).

9 Limitations

The series of studies reported here should be interpreted in the context of known limitations. As the majority of the subject population was drawn from the student body of a university, the age and nationality of the participants may not accurately

represent the U.S. population; however, as the users of social media are traditionally younger, this bias may be somewhat mitigated. Second, given the predominance of mathematically-oriented study at the university from which the subjects were drawn, bias potentially arises from this area as well. Current research in this area is examining "real" social media users in order to generalize beyond university students.

10 Future Work

Other artifacts unrelated to the structural properties of the network have also appeared through the free responses providing further insight into sources that participants use to estimate credibility, such as the attribution of credibility based on response order. ("he is first one to comment and people usually comment first if they have good idea about a particular thing").

Additional network properties of interest are the amount of information that a commenter provides (for example, the effect of including a picture with a comment, whether relevant to the conversation or not) and other linguistic cues that are likely specific to social media, such as the amount of informality, and cues to deception [35, 36]. In the future, we will continue to evaluate the combinatorial effect of information by including profile information in our experimental design. Finally, further studies will explore whether the use of these properties are unique to specific types of social media and as compared to traditional communication mediums, as this new form of communication creates inherently different user expectations from that of other mediums.

11 Conclusions

This work explores the use of social network properties as a basis for determining credibility. Our results serve as a baseline for ongoing and future studies toward estimating social qualities where there is little or no social context. These findings will be combined with other work toward creating automated classification methods for characterizing credibility in social media. We intend this research to be part of a larger body of research, e.g., [37], aimed toward creating approaches that intelligently utilize social media data as a reliable "sensor" for detecting and understanding human behavior.

Acknowledgments This material is based on work supported in part by the Defense Advanced Research Projects Agency (DARPA) under Contract No. W911NF-12-1-0043. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or the U.S. Government.

References

1. Granovetter M (1983) The strength of weak ties: a network theory revisited. Sociol Theory 1:201–233
2. Easley D, Kleinberg J (2010) Networks, crowds, and markets: reasoning about a highly connected world. Cambridge University Press, Cambridge
3. Gilbert E, Karahalios K (2009) Predicting tie strength with social media. In: Proceedings of the SIGCHI conference on human factors in computing systems, ACM
4. Antin J, Churchill EF (2011) Badges in social media: a social psychological perspective. In: CHI 2011, Vancouver, 7–12 May 2011
5. Rubin VL, Liddy ED (2006) Assessing credibility of weblogs. In: AAAI symposium on computational approaches to analysing weblogs (AAAI-CAAW), ACM, 2006
6. Berger-Wolf TY, Saia J (2006) A framework for analysis of dynamic social networks. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, ACM
7. Armstrong CL, McAdams MJ (2009) Blogs of information: how gender cues and individual motivations influence perceptions of credibility. J Comput Mediat Commun 14(3):435–456
8. Lesani M, Montazeri N (2009) Fuzzy trust aggregation and personalized trust inference in virtual social networks. Comput Intell 25(2):5183
9. Kim Y, Song H (2011) Strategies for predicting local trust based on trust propagation in social networks. Knowl-Based Syst 2:1360–1371
10. Guha R, Kumar R, Raghavan R, Tomkins A (2004) Propagation of trust and distrust. In: Proceedings of the 13th international conference on world wide web, New York
11. Brashears ME (2013) Humans use compression heuristics to improve the recall of social networks. Scientific reports, vol 3. doi:10.1038/srep01513
12. Dunbar R (2010) How many friends does one person need?: Dunbar's number and other evolutionary quirks. Faber and Faber, London
13. Page L, Brin S, Motwani R, Winograd T (1998) The PageRank citation ranking: bringing order to the web, Stanford Digital Library Technologies Project, Technical report, CA
14. Golbeck JA (2005) Computing and applying trust in web-based social networks. Ph.D. Dissertation, Department of Computer Science, University of Maryland, College Park
15. Golbeck JA, Hendler J (2006) FilmTrust: movie recommendations using trust in web-based social networks. In: Proceedings of the IEEE consumer communications and networking conference. Las Vegas
16. Castillo C, Mendoza M, Poblete B (2011) Information credibility on twitter. In: Proceedings of the 20th international conference on world wide web, ACM
17. Hutto CJ, Gilbert E, Yardi S (2013) A longitudinal study of follow predictors on twitter. In: Proceedings of the 2013 ACM annual conference on human factors in computing systems (CHI), Paris
18. Romero DM, Galuba W, Asur S, Huberman BA (2011) Influence and passivity in social media. In: Machine learning and knowledge discovery in databases, Springer, Berlin, pp 18–33
19. Janicik GA, Larrick RP (2005) Social network schemas and the learning of incomplete networks. J Personal Soc Psychol 88:348–364
20. Luhmann N (2000) Familiarity, confidence, trust: problems and alternatives. In: Gambetta D (ed) Trust: making and breaking cooperative relations, electronic edition. Department of Sociology, University of Oxford, vol 6. pp 94–107 (Chapter 6, Trust: making and breaking cooperative relations)
21. Costa PT, McCrae RR (1985) The NEO personality inventory: manual, form S and form R. Psychological Assessment Resources, 1985
22. Goldberg L (1981) Language and individual differences: the search for universals in personality lexicons. In: Wheeler L (ed) Review of personality and social psychology. Sage, Beverly Hills
23. McCrae R, Costa P (1997) Personality trait structure as a human universal. Am Psychol 52:509–516

24. McCrae R, Costa P (1991) The NEO personality inventory: using the five-factor model in counseling. J Couns Dev 69(4):367–372
25. McCrae RR, John OP (1992) An introduction to the fivefactor model and its applications. J Personal 60(2):175–215
26. Guadagno RE, Okdie BM, Eno C (2008) Who blogs? Personality predictors of blogging. Comput Hum Behav 24(5):1993–2004
27. Barrick MR, Mount MK (1991) The big five personality dimensions and job performance: a metaanalysis. Person Psychol 44(1):1–26
28. Correa T, Hinsley A, De Zuniga H (2010) Who interacts on the Web?: the intersection of users personality and social media use. Comput Hum Behav 26(2):247–253
29. McKenna KY, Bargh JA (2000) Plan 9 from cyberspace: the implications of the internet for personality and social psychology. Personal Soc Psychol Rev 4(1):57–75
30. Bachrach Y, Kosinski M, Graepel T, Kohli P, Stillwell D (2012) Personality and patterns of facebook usage. In: Proceedings of the 3rd annual ACM web science conference
31. Janicik G, Larrick T (2005) Social network schemas and the learning of incomplete networks. J Personal Soc Psychol 88:348–364
32. Yin X, Han J, Yu PS (2008) Truth discovery with multiple conflicting information providers on the web. IEEE Trans Knowl Data Eng 20:796–808
33. Kim Y, Song H (2011) Strategies for predicting local trust based on trust propagation in social networks. Knowl-Based Syst 2:1360–1371
34. Appling DS, Briscoe EJ, Hayes H, Mappus RL (2013) Towards automated personality identification using speech Acts. In: Workshop on computational personality recognition. International AAAI conference on weblogs and social media (ICWSM)
35. DePaulo BM, Lindsay JJ, Malone BE, Muhlenbruck L, Charlton K, Cooper H (2003) Cues to deception. Psychol bull 129(1):74
36. Zhou L, Burgoon JK, Nunamaker JF, Twitchell D (2004) Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications. Group Decis Negot 13(1):81–106
37. Weiss L, Briscoe E, Hayes H, Kemenova O, Harbert S, Li L, Lebanon G, Stewart C, Miller D, Foy D (2013) A comparative study of social media and traditional polling in the Egyptian uprising of 2011. In: Social computing. Behavioral-cultural modeling and prediction lecture notes in computer science, Springer, Berlin, pp 303–310

Anonymizing Social Network Data for Maximal Frequent-Sharing Pattern Mining

Benjamin C.M. Fung, Yan'an Jin, Jiaming Li and Junqiang Liu

Abstract Social network data provide valuable information for companies to better understand the characteristics of their potential customers with respect to their communities. Yet, sharing social network data in its raw form raises serious privacy concerns because a successful privacy attack not only compromises the sensitive information of the target victim but also divulges the relationship with his/her friends or even their private information. In recent years, several anonymization techniques have been proposed to solve these issues. Most of them focus on how to achieve a given privacy model but fail to preserve the data mining knowledge required for data recipients. In this paper, we propose a method to k-anonymize a social network dataset with the goal of preserving *frequent sharing patterns* and *maximal frequent sharing patterns*, the most important kinds of knowledge required for marketing and consumer behavior analysis. Experimental results on real-life data illustrate the trade-off between privacy and utility loss with respect to the preservation of (maximal) frequent sharing patterns.

Keywords Privacy protection · Anonymization · Neighborhood attack · Data mining · Frequent sharing pattern

B.C.M. Fung (✉)
McGill University, Montreal, QC, Canada
e-mail: ben.fung@mcgill.ca

Y. Jin
Huazhong University of Science and Technology, Hubei University of Economics, Hubei, China
e-mail: yan.an.jin@hbue.edu.cn

J. Li
IBM Canada Software Lab, Toronto, ON, Canada
e-mail: jiamingl@ca.ibm.com

J. Liu
Zhejiang Gongshang University, Zhejiang, China
e-mail: jjliu@alumni.sfu.ca

Ö. Ulusoy et al. (eds.), *Recommendation and Search in Social Networks*,
Lecture Notes in Social Networks, DOI 10.1007/978-3-319-14379-8_5

1 Introduction

In recent years, the emergence of social network applications such as Facebook, Twitter, and MySpace has provided a new source of information for consumer behavior analysis. By identifying the common preferences with respect to the customers' background information and their connections, a company can better customize their products and marketing strategy for different communities. Thus, there is an urge to share social network data together with the set-valued data of the participants. The set-valued data, for example, can be online purchasing transactions or click history on advertisements on social network websites. However, releasing social network data in its raw form raises serious privacy concerns. In this paper, we present a method to anonymize the social network with the goals of hiding the identities of the participants and preserving the frequent sharing patterns within a community.

1.1 Motivating Scenario

Figure 1a depicts a typical social network of 11 participants together with their names, jobs, and purchased items via the advertisements in the social network. The social network service provider wants to share such useful data to its cooperative partners who placed advertisements for market analysis. Yet, sharing such information would compromise the privacy of participants, which in turn damages the image of the social network service provider. A naive method is to de-identify the social network data by simply removing the explicit identifiers, such as name and date of birth. However, many previous works [11, 24] have already shown that simply removing explicit identifiers is insufficient because an adversary may utilize some external knowledge to identify an individual from the data. The following example illustrates a privacy attack on a de-identified social network.

Example 1 Consider the social network in Fig. 1a. Even if the names of the participants have been removed before releasing the data, an adversary may still identify an individual using *neighborhood attack* [36]. Suppose the adversary knows the target victim *Toby* has four friends and two of his friends know each other. Given such background knowledge, the adversary can easily identify *Toby*'s vertex from the social network. One effective way to thwart this kind of neighborhood attack is to ensure that the 1-neighborhood network structure of *Toby* is isomorphically similar to the 1-neighborhood network structure of at least $k-1$ other vertices in the shared social network data. This privacy model is known as k-anonymity on social network data [27, 36]. To make *Toby* 2-anonymous, two edges, indicated by the dashed lines in Fig. 1b, are added between *Ben* and *Sarah*, and *Yang* and *Sarah*. □

Achieving k-anonymity on social network is not a new problem. It has been previously studied in [36]. The challenge addressed in this paper is how to preserve the frequent patterns shared within a community [9], known as *frequent sharing*

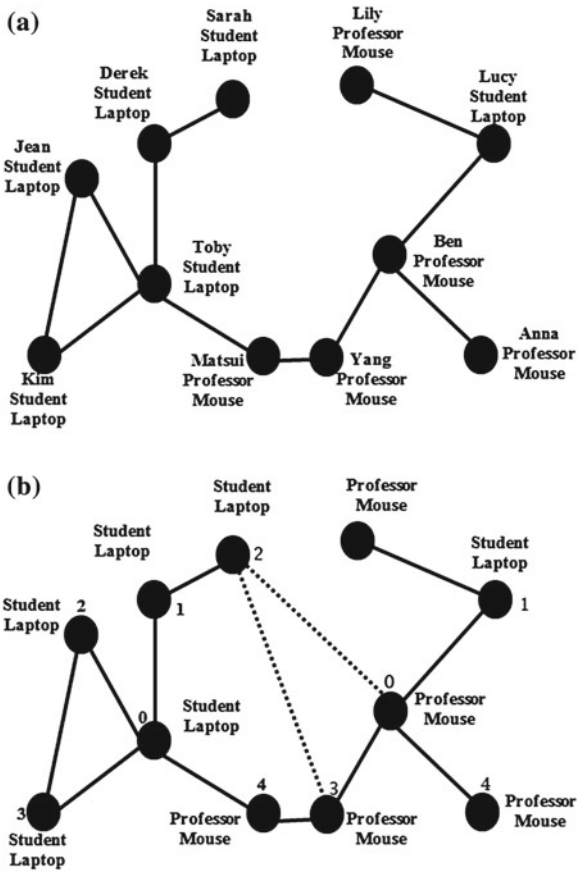

Fig. 1 Sample social network. **a** Raw social network. **b** Anonymized social network

patterns and *maximal frequent sharing patterns*, so that the data recipient can still retrieve them even from the k-anonymous social network data. Specifically, a frequent sharing pattern is a combination of vertex labels that is shared within a connected subgraph with a minimum number of vertices specified by the social network data holder. A frequent sharing pattern is maximal if any of its proper superset is not frequent. The following example illustrates the general idea of minimizing the impact on frequent sharing patterns. A formal definition is given in Sect. 3.

Example 2 Consider Fig. 1a again. The pattern {*Laptop*} has support 5 because a maximum of five connected vertices contain *Laptop*. Similarly, the pattern {*Mouse*} has support 4. Let the data-holder-specified minimum support be 5. Then {*Student*} and {*Laptop*} are frequent but {*Professor*} and {*Mouse*} are not. Though {*Student*} and {*Laptop*} are frequent sharing patterns, they are not maximal frequent sharing pattern because there exists a proper superset, namely {*Student*, *Laptop*}, that is also frequent. To make *Toby*'s neighborhood structure 2-anonymous, we have the option to add two edges as described in Example 1 or add an edge between *Lily* and *Ben*.

The latter option is less desirable because adding an edge between *Lily* and *Ben* would increase the support of {*Mouse*} and {*Professor*, *Mouse*} from 4 to 5, resulting in some false (maximal) frequent sharing patterns. □

1.2 Challenges and Contributions

The challenges of anonymizing social network data for (maximal) frequent sharing patterns mining are summarized as follows. First, social network data are a composition of graph data and set-valued data, representing the relationships among the participants and the (sensitive) personal information of the participants, respectively. Thus, existing anonymization methods for k-anonymity [26], ℓ-diversity [21], and confidence bounding [28] that are designed for tabular data are not applicable to social network data. Second, in order to preserve the (maximal) frequent sharing patterns, a straightforward approach is to first extract all (maximal) frequent sharing patterns and then minimize the impact on the extracted patterns in the anonymization process. However, the preprocessing step of extracting the (maximal) frequent sharing patterns from social network is expensive. Third, real-life social network data are usually very large; therefore, it is essential to develop a scalable anonymization algorithm.

The contributions of this paper are summarized as follows. First, to the best of our knowledge, this is the first anonymization algorithm to achieve k-anonymity on social network data while minimizing the impact on (maximal) frequent sharing patterns in the set-valued data. Achieving k-anonymity on social network data is NP-hard [36], so we present a heuristic approach to address the problem. Second, our proposed method is not only effective but also efficient to anonymize a large volume of social network data. Third, we verify the effectiveness of our proposed method by extensive experiments on real-life data. The results suggest that our algorithm can effectively preserve the privacy with reasonable trade-off between privacy and information utility measured in terms of preserving (maximal) frequent sharing patterns. The preliminary version of this paper was published in [10].

The rest of the paper is organized as follows. Related works are discussed in Sect. 2. In Sect. 3, we formally define the problem. Our proposed anonymization method for preserving (maximal) frequent sharing patterns is presented in Sect. 4. Our experimental results are shown in Sect. 5. Section 6 concludes the paper.

2 Related Work

Privacy-preserving data sharing is a broad research area. Many works have been published in the last decade. In this section, we review the related works in social network data anonymization. Social network data can be broadly categorized into several data models [12]. The simplest model is to represent the social network data as a graph, in which vertices represent the participants and edges represent their relationships.

An enhanced model is to represent social network with labeled vertices, representing the associated information of the participants, such as jobs and purchased items. In addition to labeled vertices, the third model assumes labeled edges to indicate different types of relationships. Privacy threats on social network data can be summarized into three types, namely *identity disclosure*, *attribute disclosure*, and *link re-identification*, depending on the adversary's background knowledge on the social network data. For identity disclosure, the attack goal is to identify the vertex that represents a target victim. For attribute disclosure, the attack goal is to identify or infer some sensitive information about a target victim. For link re-identification, the attack goal is to identify sensitive relationships of a target victim. This paper focuses on thwarting identity disclosures for social network with labeled vertices.

Much effort has been made for privacy preservation in social networks. Backstrom et al. [3] is the pioneer to investigate this problem. The authors argue that the naive de-identification approach does not ensure the privacy protection in both active and passive attacks. In active attacks, an adversary may succeed in re-identifying vertices and edges in the published social networks using the prior knowledge of his previously inserted subgraph. In passive attacks, an adversary may infer the identity of vertices and edges or other information in the published social network by using a special and unique subgraph which is simple to identify. However, the paper does not propose any solutions for both attacks. Korolova et al. [17] introduced a potential attack in which the adversary gathers the information of neighborhood and determines the information of the whole social network. The adversary may succeed in inferring the link structure of the global social network by gaining the neighborhood information together. Existing anonymization techniques which have been published in recent years are primarily classified into three categories: adding edges, generalization, and randomization [30]. In the following, we summarize each category and compare the differences between their techniques and ours.

Adding edges techniques on social network. Much work has been done on the basis of the classic k-anonymity model [25–27], which has been intensively studied for relational data. These techniques achieve anonymization by adding edges. Liu and Terzi [20] presented a k-degree method for anonymizing a graph by inserting and removing edges so that vertices cannot be distinguished by degrees. Wu et al. [29] proposed a k-symmetry model, which assures that any vertices in a naively anonymized network has at least $k-1$ structurally equivalent counterparts. Zhou and Pei [36] generalized node labels and inserted edges into the network to achieve k-neighborhood. Zou et al. [37] designed a k-automorphism approach, which ensures that the adversary cannot distinguish any vertex from other $k-1$ symmetric vertices based on structural information. Cheng et al. [7] introduced a k-isomorphism technique to thwart structural attacks in social networks, ensuring that even if the adversary has some information about an individual, or the relationships among the individuals, privacy will still be protected. Bonchi et al. [4] described a k-obfuscation model which ensures that an adversary cannot infer the vertex in the obfuscated graph based on the vertex of its original graph. The aforementioned approaches employ traditional utility measures, such as graph topological properties, graph spectral properties, and aggregate network queries, to evaluate the information utility of the anonymized

social network. In contrast, our proposed method aims at achieving k-anonymity on social networks with the goal of preserving (maximal) frequent itemsets in the anonymization process. In terms of privacy model, the closest work is [36]; however, their method employs node label generalization and, therefore, cannot preserve frequent sharing patterns on the node labels. There is no common ground for a fair comparison by experiments.

Generalization techniques on social network. Generalization method is also a promising privacy protection technique, which has been widely adopted in relational data anonymization. In social network data, it is still meaningful to analyze the properties of the original graph and generalized graph. Hay et al. [15] proposed an anonymization technique to divide vertices into clusters and publish a generalized graph by partitioning the graph where the privacy of any individual is properly hidden. Campan and Truta [6] provided a greedy algorithm to generalize the edges for anonymization. Generalization technique is not applicable to the problem of preserving frequent sharing patterns studied in this paper because the generalized graph is a transformation of the original graph. Therefore, it is inappropriate to compare the utility with their method in terms of frequent sharing patterns.

Randomization techniques on social network. Randomization is another commonly used approach for privacy protection. Agrawal et al. [1, 2] introduced a randomization technique by adding noise in numerical data. Enlightened by the randomization approach on numerical data, some papers exploited edges randomization tactics to anonymize the social network. Hey et al. [16] anonymized a graph by random perturbation. Zhang and Zhang [34] achieved edge anonymity in graphs by randomly adding, deleting, and swapping edges. Ying and Wu [32] proposed a spectrum of randomization strategies to preserve privacy using the edge-based graph randomization methods. Ying and Wu [33] also presented a method to generate synthetic graphs from the original graph in order to preserve privacy while keeping information utility such as maintaining shortest distance and transitivity. Wu et al. [19] introduced a low-rank approximation-based reconstruction algorithm to protect the privacy in social network by hiding feature values. Hanhijarvi et al. [14] designed an algorithm based on randomization techniques that generate a graph with a similar feature of original graph. Bonchi et al. [4] presented a random sparsification technique, in which the algorithm only randomly removes edges without adding edges. These randomization techniques are also not suitable for preserving the property of frequent sharing patterns.

3 The Problem

In this paper, we consider a social network as an undirected, unweighted graph $G = (V, E, L)$, where V represents a set of vertices, $E \subseteq V \times V$ is a set of edges without labels, L denotes a set of *categorical labels* or simply *labels* on V. $L(v) \subseteq L$ denotes a set of labels of a vertex $v \in V$. For example in Fig. 1a, $L(v_{Toby}) = \{Student, Laptop\}$

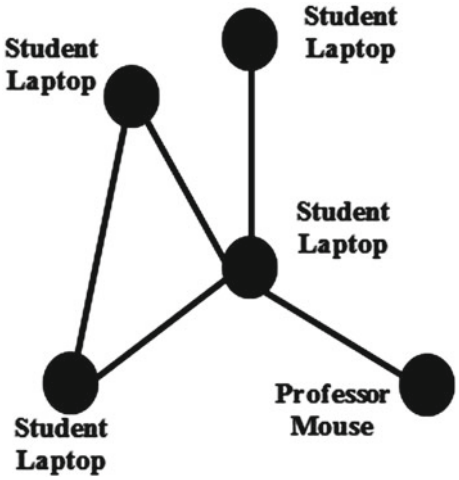

Fig. 2 1-neighborhood of *Toby*

and $L(v_{Lily}) = \{Professor, Mouse\}$. The *1-neighborhood* of a vertex v, denoted by $N^1(v)$, is the induced subgraph of the neighbors of v. For example, Fig. 2 depicts the 1-neighborhood of *Toby*, i.e., $N^1(v_{Toby})$.

The research problem studied in this paper is to transform a given social network G with labeled vertices into a k-anonymous version while preserving as many (maximal) frequent spatterns as possible. The notions of k-anonymity, frequent spatterns, and maximal frequent spatterns are formally defined as follows.

3.1 Privacy Model

Suppose an adversary knows the 1-neighborhood network structure of a target victim as background knowledge, and wants to identify the vertex of the target victim in G. To thwart this identity attack, we employ the privacy model of k-anonymity on social network [36]. The general idea is to ensure that the 1-neighborhood network structure of any vertex in a social network G is isomorphic to the 1-neighborhood network structure of at least $k - 1$ other vertices in G.

Definition 1 (*k-anonymous social network*) Let G be a social network. Let k be a privacy threshold specified by social network data holder. A vertex v in G is *k-anonymous* if there exists at least $k - 1$ other vertices $u_1, \ldots, u_{k-1} \in V$ such that $N^1(v)$ and $N^1(u_1), \ldots, N^1(u_{k-1})$ are isomorphic. A social network G is k-anonymous if every vertex $v \in V$ in G is k-anonymous [36].

For example, the social network in Fig. 1b satisfies 2-anonymity.

3.2 Frequent Spatterns

Consider a social network $G = (V, E, L)$ as defined above. Below, we formally define the notions of *sharing pattern* (*spattern*), *maximal subgraph*, *frequent spattern*, and *maximal frequent spattern* [9].

Definition 2 (*Spattern*) A *sharing pattern*, or simply *spattern*, p is a nonempty set of labels, $p \subseteq L$ and $p \neq \emptyset$. A vertex $v \in V$ *contains* a pattern p if $p \subseteq L(v)$.

To determine the popularity of a pattern within a community, we define the notion of maximal subgraph of a spattern.

Definition 3 (*Maximal subgraph of spattern*) A connected subgraph G_s of G is a *maximal subgraph of a spattern* p, denoted by $G_s(p)$, if $\forall v \forall u((v \in G_s \rightarrow p \subseteq L(v)) \wedge (u \in N^1(v) \wedge u \notin G_s \rightarrow p \nsubseteq L(u)))$. The *support* of spattern p in G_s, denoted by $Sup(p|G_s)$, is the number of vertices in G_s containing p.

The first condition $v \in G_s \rightarrow p \subseteq L(v)$ states that all vertices in G_s contain the pattern p. The second condition $u \in N^1(v) \wedge u \notin G_s \rightarrow p \nsubseteq L(u)$ states that the subgraph containing the pattern p is maximal.

Example 3 Consider Fig. 1a. Vertex v_{Toby} contains spatterns {*Student*}, {*Laptop*}, and {*Student*, *Laptop*}. Figure 3 depicts the two maximal subgraphs G_1 (composed by v_0, v_1, v_2, v_3, and v_4) and G_2 (composed by v_6) of spattern {*Student*, *Laptop*}. $Sup(\{Student, Laptop\}|G_1) = 5$ and $Sup(\{Student, Laptop\}|G_2) = 1$. G_3(composed by v_0, v_1, v_2, and v_3) is not a maximal subgraph of a spattern since v_4 is connected to G_3, meanwhile, v_4 and G_3 have the same spattern {*Student*, *Laptop*}. □

Next, we define the notion of *frequent spattern*, which captures the items that occur together frequently within a community. In case a spattern occurs frequently in multiple groups, its support is represented by the maximum support among all the groups.

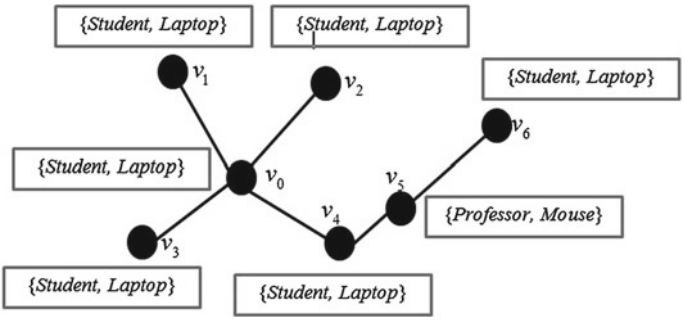

Fig. 3 Example of maximal subgraph of a spattern

Definition 4 (*Frequent spattern*) Let $G_1(p), \ldots, G_m(p)$ be all the maximal subgraphs of a spattern p in G. The *support* of a spattern p in G, denoted by $Sup(p)$, is the $max(Sup(p|G_1(p)), \ldots, Sup(p|G_m(p)))$. Let *MinSup* be the minimum support threshold specified by the social network data holder. A spattern p is a frequent spattern in G if $Sup(p) \geq MinSup$.

Though frequent spatterns can effectively capture the items that occur together frequently within a community, the number of frequent spatterns can be very large, and may overwhelm a user. Thus, we introduce the notion of maximal frequent spattern that captures the maximal set of items that occur together frequently in a community. Given a maximal frequent spattern, a user can derive all the frequent spatterns by identifying all the subsets of the maximal frequent spattern. Yet, the support counts of derived frequent spatterns will be lost.

Definition 5 (*Maximal frequent spattern*) A frequent spattern is a *maximal frequent spattern* in G if any of its proper superset is not frequent in G.

Example 4 Consider Fig. 1b with the additional edges. Suppose $MinSup = 5$. {*Laptop*}, {*Student*}, and {*Student*, *Laptop*} are frequent spatterns. {*Student*, *Laptop*} is a maximal frequent spattern. {*Professor*}, {*Mouse*}, and {*Professor*, *Mouse*} have support 4, so they are neither frequent spatterns nor maximal frequent spatterns. □

3.3 Problem Statement

In this paper, we addressed two research problems.

Definition 6 (*Social Network Anonymization for Frequent Spatterns*) Given a social network G with labeled vertices, a k-anonymity requirement, and a minimum support threshold *MinSup*, the *problem of anonymization of social network for frequent spatterns* is to transform G to satisfy the given k-anonymity requirement while preserving as many frequent spatterns as possible.

Definition 7 (*Social Network Anonymization for Maximal Frequent Spatterns*) Given a social network G with labeled vertices, a k-anonymity requirement, and a minimum support threshold *MinSup*, the *problem of anonymization of social network for maximal frequent spatterns* is to transform G to satisfy the given k-anonymity requirement while preserving as many maximal frequent spatterns as possible.

Achieving k-anonymity in a social network has been proven to be NP-hard [36]. Thus, we propose a heuristic approach to tackle the two aforementioned problems.

4 The Anonymization Method

In this section, we present a method to anonymize the social network $G = (V, E, L)$ to achieve k-anonymity while preserving the (maximal) frequent spatterns as described in Definitions 6 and 7. Algorithm 1 provides an overview of the

Algorithm 1 Overview of the Anonymization Algorithm

Input: Social network $G = (V, E, L)$ and anonymization threshold k;
Output: k-anonymous social network;
1: *VList* $\leftarrow V$;
2: Sort *VList* by degrees in descending order;
3: **while** *VList* $\neq \emptyset$ **do**
4: *TopK* $\leftarrow$ first k disjointed vertices in *VList*;
5: Call *SmoothingDegree*(*TopK*);
6: Call *MakeIsomorphic*(*TopK*, *AffectedV*);
7: *VList.Remove*(*TopK*);
8: *VList.InsertAndSort*(*AffectedV*);
9: **end while**

algorithm. According to the power law degree distribution [8], most of the vertices in a social network have low degrees, and only few vertices have large degrees. Therefore, our proposed method starts anonymization from the vertices with the largest degrees. The vertices V are first sorted in descending order by their degrees, and the sorted vertices are stored into *VList*. Then the algorithm iteratively processes the first k disjointed vertices in *VList*, denoted by *TopK*, and removes *TopK* from *VList*. At each iteration, two steps are performed to process *TopK* vertices. The first step is to transform the *TopK* vertices to have the same degree. The second step is to extract the 1-neighborhood of the *TopK* vertices and add edges to make them isomorphic. The challenge is that making a group of vertices isomorphic may break the isomorphism of some previously processed vertices. Thus, the algorithm has to add the affected vertices, denoted by *AffectedV*, back to *VList*. This process repeats until *VList* becomes empty. The details of the two steps, namely *SmoothingDegree* (Line 5) and *MakeIsomorphic* (Line 6), are described as follows:

4.1 Degree Smoothing

Given k disjointed vertices, denoted by *TopK*, that are sorted by degree in descending order, this step aims at making them to have the same degree by adding edges. Algorithm 2 describes the general idea of this procedure. Let v_0 be the first vertex of *TopK*, i.e., the one with the largest degree among the k vertices. For each vertex v_i in *TopK*, the procedure counts the number of edges, denoted by d, required to be added to v_i, and heuristically selects d vertices with the least degrees from V. Vertices with low degrees are preferable because they can be efficiently obtained from the end of the *VList*, and they are relatively easy to smoothen, if necessary, in later iterations. Due to the power law degree distribution [8], it is very likely that more than d vertices have the same least degree. The question is how to select the vertices from these candidates for adding edges with minimal impacts on the (maximal) frequent spatterns.

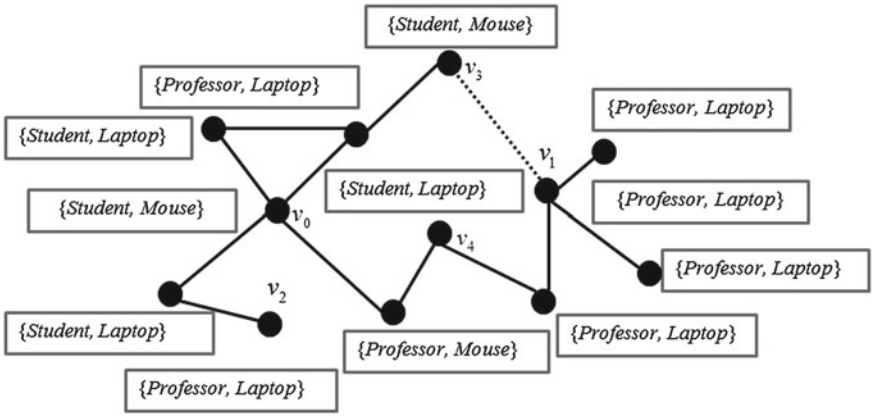

Fig. 4 Degree smoothing with $k = 2$

Adding edges increases the support of some spatterns. Consequently, some spatterns that were not (maximal) frequent before the anonymization may become (maximal) frequent after the anonymization, resulting in some false (maximal) frequent spatterns. Thus, the heuristic function for selecting the target vertices should minimize the increase of the support. In other words, the function selects a vertex v_j with a label that has minimal overlap with the label of vertex v_i:

$$Cost(v_i, v_j) = |L(v_i) \cap L(v_j)| \tag{1}$$

where $L(v_i)$ and $L(v_j)$ denote the labels of v_i and v_j, respectively. If more than d vertices share the same least degree and *Cost*, the algorithm randomly chooses d of them.

Example 5 Consider Fig. 4 with $k = 2$. After sorting all vertices by degree descending order, v_0 has the largest $degree(v_0) = 4$, v_1 is the vertex with the second largest degree, with $degree(v_1) = 3$, that is not connected with v_0. Thus, $d = degree(v_0) - degree(v_1) = 1$, and one edge has to be added between v_1 and an another vertex, which has the least degree. In this example, both v_2 and v_3 have degree 1; therefore, we choose the one with minimum overlap in their labels: $Cost(v_1, v_2) = |\{Professor, Laptop\} \cap \{Professor, Laptop\}| = 2$ and $Cost(v_1, v_3) = |\{Professor, Laptop\} \cap \{Student, Mouse\}| = 0$. Since $Cost(v_1, v_3) < Cost(v_1, v_2)$, we add an edge between v_1 and v_3. □

4.2 Making Isomorphic

After smoothing the degree of the *TopK* vertices, the next step is to make them isomorphic by adding edges to the 1-neighborhood of *TopK* vertices. Similar to *DFS Code* [31], we employ a technique called *BFS coding* to identify the missing

Algorithm 2 *SmoothingDegree*(*TopK*)

Input: *TopK* sorted by degrees in descending order;
Output: *TopK* with the same degree;
1: $v_0 = TopK.popfirst()$;
2: **while** $TopK \neq \emptyset$ **do**
3: $v_i = TopK.popfirst()$;
4: $d = degree(v_0) - degree(v_i)$;
5: Add d edges to v_i based on minimum *Cost*;
6: **end while**

edges. Algorithm 3 describes the detailed steps. The general idea is to compare the 1-neighborhood of the first vertex, denoted by $N^1(v_0)$, with the 1-neighborhood of each of the remaining vertices, denoted by $N^1(v_x)$, in *TopK*, and compare their BFS codes to add the missing edges (Lines 5–6). The next task is to identify the previously k-anonymized vertices that are ruined by the newly added edges. In other words, these affected vertices, denoted by *AffectedV*, have to be put back to the *VList* for re-anonymization. Lines 7–16 describe this detection process. A previously k-anonymized vertex is affected by the newly added edges if it satisfies one of the following conditions:

1. the vertex is a neighbor of v_0 or a neighbor of v_x (Line 8), or
2. the vertex is in *TopK* and shares the same k-anonymous group with another vertex v_a such that v_a is a neighbor of v_0 or a neighbor of v_x (Lines 10–14).

After the first round, the 1-neighborhood of v_0 is the supergraph of others. Then the algorithm runs the same steps once again to ensure that the structure of the 1-neighborhood of all vertices in *TopK* are copies of the 1-neighborhood of v_0. In the rest of this section, we focus on how to compute the BFS code of the 1-neighborhood of a given vertex, and how to compute two BFS codes in order to determine the missing edges.

To facilitate the comparison of the structure of graphs, we use a *breath-first search tree* (*BFS tree*) to encode the two graphs and compare their BFS codes. The general idea is to traverse the vertices using a breath-first search by following the subscripts of the vertices. Consider Fig. 5 as an example. We start the BFS coding from the vertex with the largest degree, which is v_0, followed by v_1, v_2, v_3, and finally the edges between v_2 and v_3. Thus, The 1-neighborhood BFS Code of v_0 denoted by $BFS(N^1(v_0))$, is (01020323).

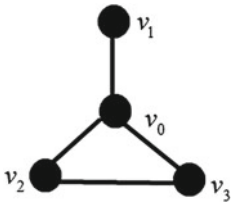

Fig. 5 BFS tree

Algorithm 3 *MakeIsomorphic*(*TopK*, *AffectedV*)

Input: *TopK* sorted by degrees in descending order;
Output: *TopK* with isomorphic 1-neighborhood;

```
1: v0 = TopK.popfirst();
2: for i := 1 to 2 do
3:   for each vx ∈ TopK do
4:     if BFS(N1(v0)) ≠ BFS(N1(vx)) then
5:       Add edges to N1(v0) based on BFS(N1(vx));
6:       Add edges to N1(vx) based on BFS(N1(v0));
7:       for each va ∉ VList do
8:         if va ∈ N1(v0) ∨ va ∈ N1(vx) then
9:           AffectedV.Add(va);
10:            for each vy ∈ TopK do
11:              if vy ∈ AnonymousGroup(va) then
12:                AffectedV.Add(vy);
13:              end if
14:            end for
15:          end if
16:        end for
17:      end if
18:    end for
19: end for
```

Next, we can determine the missing edges between two subgraphs by comparing their BFS codes. The following example illustrates the idea.

Example 6 Consider Fig. 6. The encoding always starts from 0, so v_5–v_8 in Fig. 6b become v_0–v_4. The BFS Codes of $N^1(v_0)$ and $N^1(v_5)$ are (010203041423) and (0102030434), respectively. By comparing the two BFS codes, we know that (14) and (23) are not in $N^1(v_5)$, and (34) is not in $N^1(v_0)$. Therefore, we add an edge between v_3 and v_4 in $N^1(v_0)$ and add two edges between v_6 and v_9 and between v_7 and v_8 in $N^1(v_5)$. After adding these edges, the two graphs become isomorphic. □

The following example illustrates how to isomorphize the 1-neighborhood of three vertices.

Example 7 Consider the 1-neighborhoods of v_0, v_5, and v_{10} in Fig. 7a. To make them isomorphic, we start from $N^1(v_0)$ and iteratively compare it with $N^1(v_5)$ and

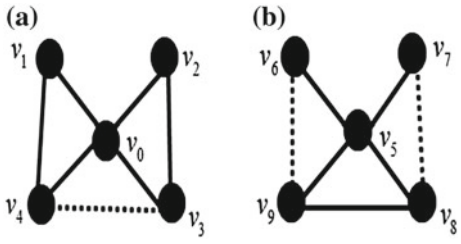

Fig. 6 Making isomorphic. **a** $N^1(v_0)$. **b** $N^1(v_5)$

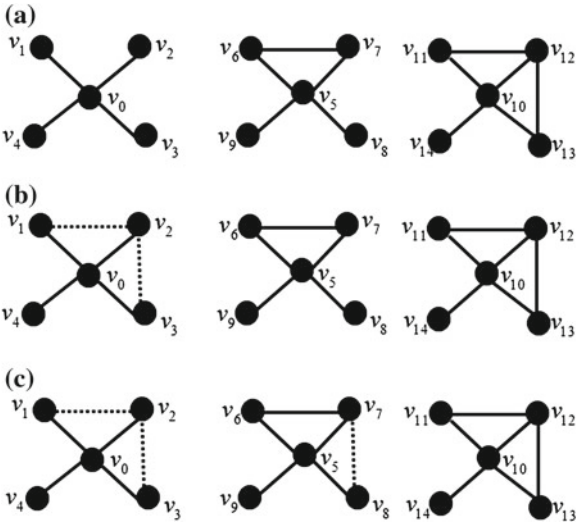

Fig. 7 Example of 3 neighborhoods anonymization. **a** 3 neighborhoods before anonymization. **b** 3 neighborhoods after first anonymization, the *dashed edges* are new added edges. **c** 3 neighborhoods after second anonymization, the *dashed edges* are new added edges

$N^1(v_{10})$. By comparing $BFS(N^1(v_0))$ with $BFS(N^1(v_5))$ and $BFS(N^1(v_{10}))$, we add an edge between v_1 and v_2 and another edge between v_2 and v_3 as shown in Fig. 7b. Yet, the three 1-neighborhoods are not isomorphic yet because $N^1(v_5)$ and $N^1(v_{10})$ are different. Since $N^1(v_0)$ must be a supergraph of $N^1(v_5)$ and $N^1(v_{10})$. We once again compare $BFS(N^1(v_0))$ with $BFS(N^1(v_5))$ and $BFS(N^1(v_{10}))$, add an edge between v_7 and v_8 as depicted in Fig. 7c. □

4.3 Analysis and Discussion

In this section, we analyze the computational complexity of the aforementioned procedures and discuss the limitations of our proposed algorithm.

In the *SmoothingDegree* algorithm, the heuristic function first selects the vertices with the lowest degree and then evaluates the impact on spatterns. The computational complexity of the algorithm is $O(kn \log n)$, where k is the anonymization threshold, n is the number of the vertices with the lowest degree. In the *MakeIsomorphic* algorithm, we use BFS code to encode the 1-neighborhood of a given vertex. We also need to find those affected vertices and put them into *VList* again. Considering the worst case, the computational complexity of the algorithm is $O(k^3 \times |V| + k \times |V|^2)$, where k is the anonymization threshold and $|V|$ is the number of vertices in $N^1(v_i)$, where $v_i \in TopK$.

An alternative solution to tackle the problem is to first extract the (maximal) frequent spatterns from the raw social network graph. Then at each iteration, the method chooses a vertex for adding edge with a heuristic function that minimizes the impact on the (maximal) frequent spatterns. This alternative solution suffers from two shortcomings:

1. Extracting (maximal) frequent spatterns is computationally expensive, and doing so will significantly increase the complexity of the anonymization algorithm.
2. The notion of (maximal) frequent spatterns depends on the user-specified minimum support threshold. In real-life data publishing, it is difficult for the data holder to determine an appropriate minimum threshold in advance on behalf of the data recipient. Also, the evaluation must then depend on the specified minimum support threshold.

Supported by the experimental results, we would like to emphasize that our proposed algorithm can effectively preserve the (maximal) frequent itemsets although the algorithm does not actually extract the (maximal) frequent itemsets from the social network.

We would also like to provide a justification on why we choose the vertices with the lowest degree in the *SmoothingDegree* algorithm. First, adding edges between vertices with large degrees may affect the previously anonymized vertices and increase the chance of re-anonymization, which degrades the efficiency and affects the diameter of the social network [36]. Second, since the BFS Coding technique can only deal with the disjointed vertices, adding edges between vertices with large degrees will corrupt the disjointed vertices and increase the difficulty to achieve k-anonymity.

The resulting anonymous social network is effective for preventing attacks that rely on neighborhood information. For example, the de-anonymization method proposed by Narayanan and Shmatikov [24] is not applicable to our k-anonymous social network. Their de-anonymization algorithm has two steps, namely seed identification and propagation. Both steps are disoriented by the k-anonymous structures. Though k-anonymity technique is effective to thwart neighborhood attack on social network, our approach also has some limitations. First, our approach can only deal with 1-neighborhood, if an adversary has the background knowledge beyond 1-neighborhood, the k-anonymous social network may still suffer from neighborhood attacks. Second, we assume that the adversary has the background knowledge of the structure of the social network. If the adversary has both the structural background knowledge of the social network and the partial label information of the target victim, our approach is insufficient for this kind of attack. Third, our proposed BFS Coding technique can only deal with disjointed vertices. We cannot guarantee that the minimal number of edges are added to achieve k-anonymity since determining whether two neighborhoods are isomorphic is NP-Hard [13] and making two neighborhoods isomorphic is also NP-Hard [36].

Example 8 Consider the 1-neighborhoods of v_0 and v_1 in Fig. 8. To make them isomorphic using BFS Coding technique, an edge will be added on v_0 and v_6 by comparing $BFS(N^1(v_0))$ with $BFS(N^1(v_1))$. However, $N^1(v_0)$ and $N^1(v_1)$ will not be isomorphic since the added edge demolishes the structure of $N^1(v_0)$. BFS Coding technique is not suitable for anonymizing the neighborhoods of connected vertices. □

Example 9 Consider the 1-neighborhoods of v_0 and v_1 in Fig. 9a. Intuitively, $N^1(v_0)$ and $N^1(v_1)$ are isomorphic before anonymization. Yet, due to the limitation of our algorithm, we have to extract $BFS(N^1(v_0))$ and $BFS(N^1(v_1))$, and add edges by

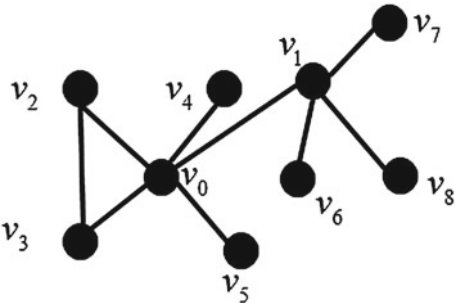

Fig. 8 BFS coding technique (neighborhoods of connected vertices)

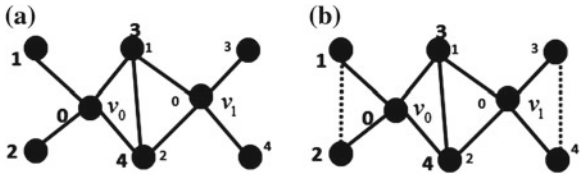

Fig. 9 BFS coding technique (adding edges) **a** before **b** after

comparing $BFS(N^1(v_0))$ with $BFS(N^1(v_1))$. The BFS Coding technique cannot guarantee adding minimum number of edges for achieving k-anonymity. □

5 Experimental Evaluation

The objective of the experiments is to evaluate the performance of the proposed algorithm with respect to the data quality of the anonymous social network, efficiency, and scalability of the anonymization process. The experiments were conducted on a PC with Core i7 2GHz CPU with 8GB memory running on Windows 7.

5.1 Datasets

We conducted the experiments on three real-life datasets, namely *Gnutella05*,[1] *Gnutella08*[2] [18], and *Adult*.[3] *Gnutella05* and *Gnutella08* are snapshots of the Gnutella peer-to-peer file sharing network in August 2002. In both datasets, vertices represent host computers and the edges represent the connections. *Gnutella05* has 8,846 vertices and 31,839 edges. *Gnutella08* has 6,301 vertices and 20,777 edges. We converted the original directed graphs into indirected graphs for our experiment.

[1] http://snap.stanford.edu/data/p2p-Gnutella05.html.

[2] http://snap.stanford.edu/data/p2p-Gnutella08.html.

[3] http://archive.ics.uci.edu/ml/datasets/Adult.

Algorithm 4 Relabel(v, *VList*)

Input: A vertex v and a list of vertices excluding v
Output: A relabeled social network
1: **for** $i = 1$ to M **do**
2: **for each** $x \in N^1(v)$ **do**
3: **if** $L_i(x) == L_i(v)$ **then**
4: $labelNum(L_i(x)) \leftarrow labelNum(L_i(v))$;
5: Relabel(x, $VList - x$);
6: **end if**
7: **end for**
8: **end for**

As the two datasets have no labels, we used the *Adult* dataset, which has been previously employed in [22, 23], to synthesize the vertex labels. *Adult* has 45,222 records on 8 categorical attributes.

As the number of records in *Adult* are different from the number of vertices in *Gnutella05* and *Gnutella08*, we sequentially associated each record in *adult* with *Gnutella05* and *Gnutella08* based on the order given in the raw datasets. The numerical attributes in *Adult* dataset were removed.

5.2 (Maximal) Frequent Spatterns Extraction

To evaluate the data utility on frequent spatterns, we measure the change of the (maximal) frequent spatterns before and after anonymization. We first briefly explain how to extract the (maximal) frequent spatterns using a tool called *MAFIA* [5], followed by the experimental results.

In frequent itemsets mining, the support of an itemset is simply the number of transactions containing the itemset. However, in frequent spatterns mining, we *cannot* simply treat the label of each vertex as a transaction because the support of a spattern is the number of vertices in a maximal subgraph of the spattern See Definition 3. In other words, even if two disjoint vertices have the same label, the support of the label is only 1. Thus, we need to first relabel the vertex labels such that two labels share the same label number only if they have the same label and their vertices are connected.

Let $L_j(v)$, a *sublabel* of $L(v)$, be the jth label of vertex v. For example, v_{Toby} has $L_1(v_{Toby}) = \{Student\}$ and $L_2(v_{Toby}) = \{Laptop\}$ in Fig. 1a. The first step is to assign a temporary distinct sublabel number to each sublabel of every vertex, denoted by $labelNum(L_i(v))$, and then call the depth-first recursive function $Relabel(v, V - v)$, where v can be any vertex in V, as described in Algorithm 4. The general idea is to iterate through each sublabel of every neighbor of a given vertex v and copy the sublabel number from v to its neighbor x if their sublabels are the same. To avoid relabeling the same sublabel more than once, we use a boolean flag to skip the visited vertices.

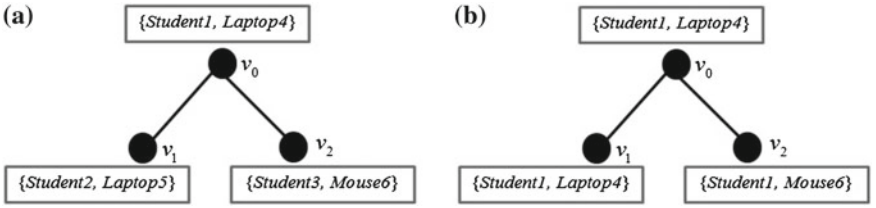

Fig. 10 Relabeling. **a** Sublabel after assignment. **b** Relabeled graph with spatterns

Example 10 Consider Fig. 10a. The label in each vertex contains two sublabels: $L_1(v)$ and $L_2(v)$. We first assign a distinct sublabel number to every sublabel. For example, $labelNum(L_1(v_2)) = 3$ and $labelNum(L_2(v_1)) = 5$. Next, we start a depth-first search on v_0 for L_1 since it has the lowest sublabel number. v_0, v_1, and v_2 are connected and $L_1(v_0)$, $L_1(v_1)$, and $L_1(v_2)$ are the same, so we reassign the sublabel numbers of *Student* in v_1 and v_2 to $labelNum(L_1(v_1)) = 1$ and $labelNum(L_1(v_2)) = 1$, respectively. Similarly, we reassign the sublabel number of *Laptop* in v_1 to $labelNum(L_2(v_1)) = 4$. Figure 10b depicts the relabeled graph. □

After relabeling the vertices, each vertex is transformed into a transaction and its sublabel numbers are treated as transaction items. Then MAFIA is applied to extract the (maximal) frequent spatterns.

5.3 Data Utility on Frequent Spatterns

The first experiment is to evaluate the impact of anonymization on frequent spatterns. The utility loss is calculated by

$$FSLoss = \frac{A - B}{B}, \tag{2}$$

where B and A denote the number of frequent spatterns extracted before and after anonymization, respectively. The value of *FSLoss* is nonnegative. The higher value of *FSLoss* means the higher number of false positive frequent spatterns, implying higher utility loss.

Figures 11 depicts the utility loss on frequent spatterns with anonymization threshold $5 \leq k \leq 20$, and minimum support $MinSup = 8, 12, 16$ and $20\,\%$ on *p2p-Gnutella08* and *p2p-Gnutella05*. For example, at $MinSup = 16\,\%$, $FSLoss = 21.3, 22.7, 26.7$ and $28\,\%$ for $5 \leq k \leq 20$, respectively. This result suggests that as k increases, more fake edges have to be added in order to meet the k-anonymity requirement, resulting in higher *FSLoss*. Yet, the impact of anonymization on *FSLoss* is mild.

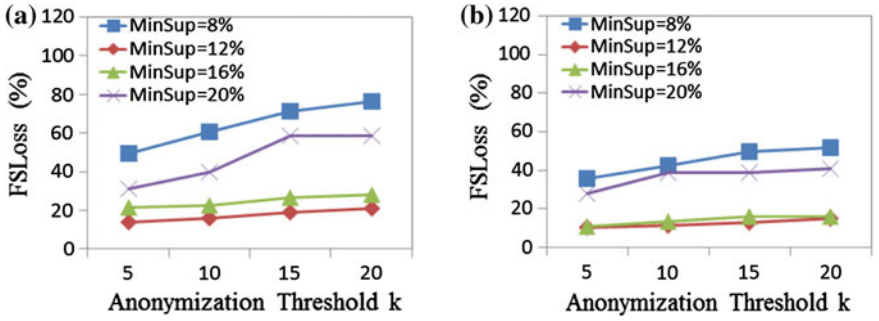

Fig. 11 Utility loss on frequent spatterns, **a** *p2p-Gnutella08*, **b** *p2p-Gnutella05*

A simple yet less effective alternative solution is to randomly choose a vertex for adding edges when we process the *TopK* vertices as discussed in Sect. 4.1. Specifically, after finding more than d vertices with the same least degree, the random method chooses d vertices and adds edges between v_1 and these d vertices. In other words, the random anonymization method does not consider the impact on spatterns on these vertices. This can be seen from Figs. 12 and 14. Figure 12a, b depict the performance of our method and this random method on both datasets. At $k = 10$ and $MinSup = 20\,\%$, our method yields $FSLoss = 38.9\,\%$ while the random method yields $FSLoss = 51.9\,\%$. Figure 12b shows similar results on *p2p-Gnutella05*. The experimental results suggest that our proposed method consistently yields lower *FSLoss* than the random method. The benefit of our method over the random method is more obvious on the smaller dataset, *p2p-Gnutella08*.

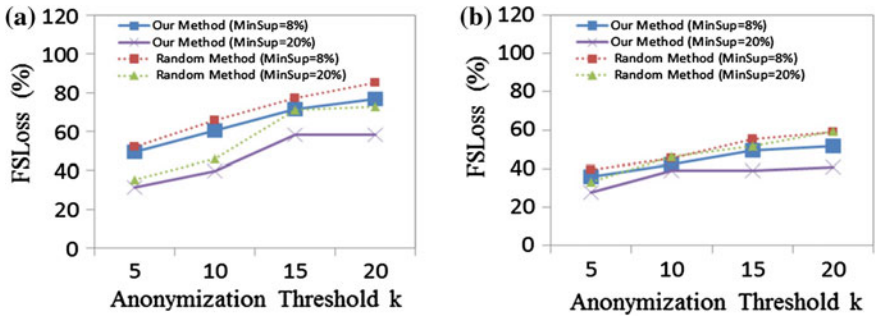

Fig. 12 Comparing with the random method with respect to *FSLoss*, **a** *p2p-Gnutella08*, **b** *p2p-Gnutella05*

5.4 Data Utility on Maximal Frequent Spatterns

To evaluate the impact on maximal frequent spatterns, we use MAFIA to extract maximal frequent spatterns before and after anonymization. Since adding edges will increase the support of maximal frequent spatterns which will cause utility loss, we measured the average increase of the support of the maximal frequent spatterns. The utility loss is calculated by

$$MFSLoss = \frac{\sum_{i=1}^{|MFS|} \frac{Sup(p_i)' - Sup(p_i)}{Sup(p_i)}}{|MFS|}, \tag{3}$$

where $Sup(p_i)$ and $Sup(p_i)'$ represent the support counts of a maximal frequent spattern p_i before and after anonymization and *MFS* denotes the number of maximal frequent spatterns before anonymization. The value of *MFSLoss* is nonnegative. The higher value of *MFSLoss* means the higher increment of the support of maximal frequent spatterns, implying higher utility loss.

Figure 13 describes the utility loss on maximal frequent spatterns with anonymization threshold $5 \leq k \leq 20$, and minimum support $MinSup = 25\,\%, 30\,\%, 35\,\%, 40\,\%$ on *p2p-Gnutella08* and *p2p-Gnutella05*, respectively. At $MinSup = 30\,\%$, *MFSLoss* spans from 7.6 to 9.6 % for $5 \leq k \leq 20$. *MFSLoss* generally increases as k increases in both datasets. At $MinSup = 35\,\%, 40\,\%$, they almost have the same *MFSLoss* on both datasets. The reason is that with the increase of minimum support, only a few maximal frequent spatterns are extracted and many of them are identical. Therefore, the utility loss is almost identical.

We also compared our method with the random method discussed in Sect. 5.3. Figure 14a, b depict the utility loss on two datasets using different anonymization method, respectively. Consider Fig. 14a. At $k = 20$, and $MinSup = 40\,\%$, our method yields $MFSLoss = 14.3\,\%$ while the random method yields $MFSLoss = 16.7\,\%$. Again, the experimental results suggest that our proposed method consistently outperforms the random method in terms of *MFSLoss*.

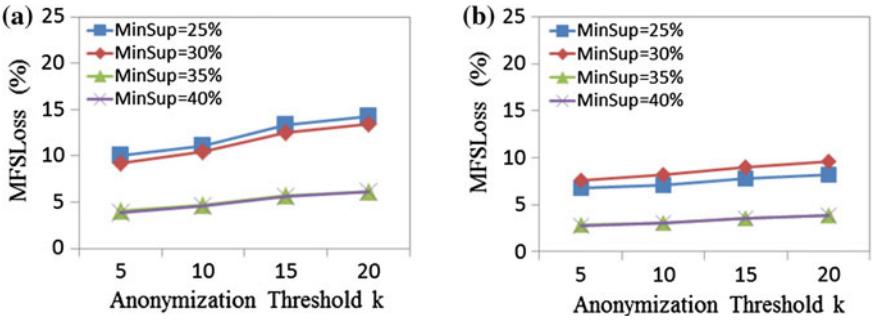

Fig. 13 Utility loss on maximal frequent spatterns, **a** *p2p-Gnutella08*, **b** *p2p-Gnutella05*

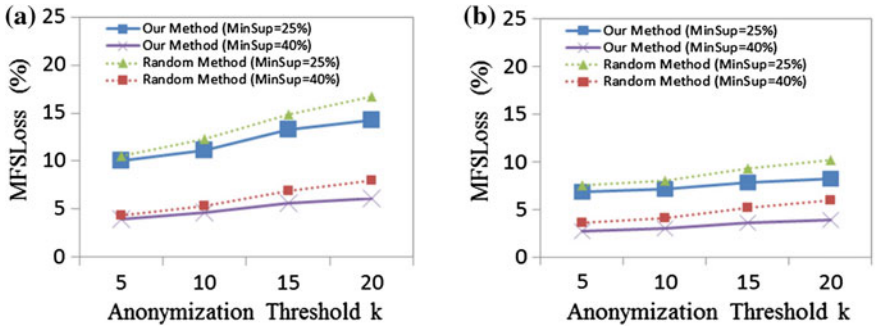

Fig. 14 Comparing with the random method with respect to *MFSLoss*, **a** *p2p-Gnutella08*, **b** *p2p-Gnutella05*

5.5 *Efficiency and Scalability*

We evaluated the efficiency of our proposed method on the two real-life datasets. Figure 15a depicts the runtime on two datasets with anonymization threshold $5 \leq k \leq 20$. At $k = 20$, the runtimes are 222 and 406 s on *p2p-Gnutella08* and *p2p-Gnutella05*, respectively. The dataset *p2p-Gnutella05* takes longer time because its size in terms of the number of vertices and edges is larger than another one. In both datasets, with the increase of k, the runtime increases because a larger k implies a more stringent privacy requirement, which in turns making it more difficult to achieve k-anonymity. Specifically, Algorithm 2 and Algorithm 3 have to add more edges in order to smoothen the degrees and achieve k-isomorphism, respectively.

We employed *p2p-Gnutella05* to evaluate the scalability of our algorithm with the first 2,000, 4,000, 6,000, and 8,000 vertices. Figure 15b depicts the runtime of the anonymization algorithm at $k = 20$. The total runtimes for anonymizing 2,000, 4,000, 6,000, and 8,000 vertices are 7.33, 55.63, 186.98, and 377.60 s, respectively. The results are consistent with the complexity analysis in Sect. 4.3. The runtime is primarily dominated by Algorithm 3 for achieving k-isomorphism.

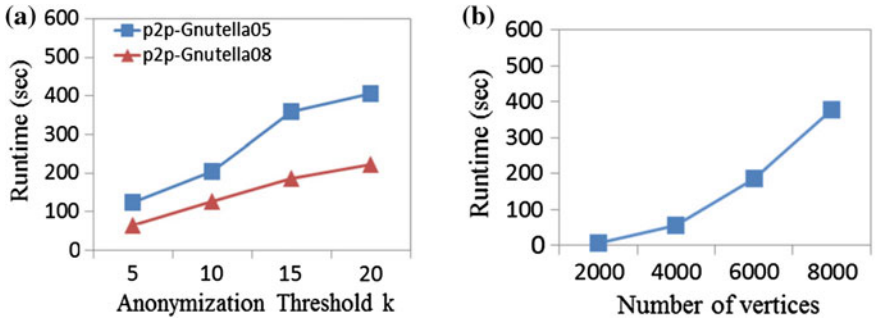

Fig. 15 Performance, **a** efficiency **b** scalability on *p2p-Gnutella05*

6 Conclusion

Social networks provide valuable information for market analysis. Market analysts can extract buying patterns or preference of some specific communities by extracting the frequent sharing patterns from their social networks. In this paper, we have formally defined the problem of anonymization of social networks for (maximal) frequent sharing patterns, and have presented an anonymization method to thwart the potential neighborhood attacks with the goal of preserving (maximal) frequent sharing patterns. Experimental results on real-life data suggest that our proposed method can effectively preserve most of the (maximal) frequent sharing patterns in the k-anonymous social network. With reasonable minimum support and privacy thresholds, our method can preserve 80 % of the frequent sharing patterns and 95 % of the maximal frequent sharing patterns. The experiments also illustrate that there is a trade-off between privacy protection and data mining utility in anonymous datasets.

In this paper, our proposed method aims at enforcing k-anonymity on social networks. The method is effective for preventing neighborhood privacy attacks, but may fail to prevent other types of privacy attacks, such as attribute disclosure and link re-identification [12]. One possible extension of this paper is to achieve other privacy models [35] while preserving the information utility for frequent sharing patterns mining.

Acknowledgments The research is supported in part by the Discovery Grants (356065-2013) from the Natural Sciences and Engineering Research Council of Canada (NSERC), the National Natural Science Foundation of China (61272306), and the Zhejiang Provincial Natural Science Foundation of China (LY12F02024).

References

1. Agrawal D, Aggarwal CC (2001) On the design and quantification of privacy preserving data mining algorithms. In: Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems, pp 247–255
2. Agrawal R, Srikant R (2000) Privacy-preserving data mining. In: Proceedings of the ACM SIGMOD international conference on management of data, pp 439–450
3. Backstrom L, Dwork C, Kleinberg J (2007) Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In: Proceedings of the 16th international conference on world wide web, pp 181–190
4. Bonchi F, Gionis A, Tassa T (2011) Identity obfuscation in graphs through the information theoretic lens. In: Proceedings of the 27th IEEE international conference on data engineering (ICDE), pp 924–935
5. Burdick D, Calimlim M, Gehrke J (2001) Mafia: a maximal frequent itemset algorithm for transactional databases. In: Proceedings of the 17th international conference on data engineering, pp 443–452
6. Campan A, Truta TM (2008) A clustering approach for data and structural anonymity in social networks. In: Proceedings of the 2nd ACM SIGKDD international workshop on privacy, security, and trust in KDD workshop, pp 1–10

7. Cheng J, Wai-Chee Fu A, Liu J (2010) K-isomorphism: privacy preserving network publication against structural attacks. In: Proceedings of the ACM SIGMOD international conference on management of data, pp 459–470
8. Faloutsos M, Faloutsos P, Faloutsos C (1999) On power-law relationships of the internet topology. In: Proceedings of the conference on applications, technologies, architectures, and protocols for computer communication, pp 251–262
9. Fukuzaki M, Seki M, Kashima H, Sese J (2010) Finding itemset-sharing patterns in a large itemset-associated graph. In Proceedings of the 14th Pacific-Asia conference on advances in knowledge discovery and data mining, pp 147–159
10. Fung BCM, Jin Y, Li J (2013). Preserving privacy and frequent sharing patterns for social network data publishing. In: Proceedings of the 5th IEEE/ACM international conference on social networks analysis and mining (ASONAM), Niagara Falls, Canada, pp 479–485
11. Fung BCM, Wang K, Chen R, Yu PS (2010) Privacy-preserving data publishing: a survey of recent developments. ACM Comput Surv 42(4):14:1–14:53
12. Fung BCM, Wang K, Wai-Chee Fu A, Yu PS (2010) Introduction to privacy-preserving data publishing: concepts and techniques. Data mining and knowledge discovery. Chapman & Hall/CRC, Boca Raton
13. Garey MR, Johnson DS (1979) Computers and intractability; a guide to the theory of NP-completeness. W. H. Freeman and Company, New York
14. Hanhijärvi S, Garriga GC, Puolamäki K (2009) Randomization techniques for graphs. In: Proceedings of the 9th SIAM international conference on data mining (SDM), pp 780–791
15. Hay M, Miklau G, Jensen D, Towsley D, Weis P (2008) Resisting structural re-identification in anonymized social networks. Proc VLDB Endow 1(1):102–114
16. Hay M, Miklau G, Jensen D, Weis P, Srivastava S (2007) Anonymizing social networks. Technical Report 07–19, Computer Science Department, University of Massachusetts Amherst
17. Korolova A, Motwani R, Nabar SU, Xu Y (2008) Link privacy in social networks. In: Proceedings of the 17th ACM Conference on information and knowledge management, pp 289–298
18. Leskovec J, Kleinberg J, Faloutsos C (2007) Graph evolution: densification and shrinking diameters. ACM Trans Knowl Discov Data (TKDD), vol 1
19. Wu XYL, Wu X (2010) Reconstruction from randomized graph via low rank approximation. In: Proceedings of the 10th SIAM international conference on data mining, pp 60–71
20. Liu K, Terzi E (2008) Towards identity anonymization graphs. In: Proceedings of the ACM SIGMOD international conference on management of data, pp 93–106
21. Machanavajjhala A, Kifer D, Gehrke J, Venkitasubramaniam M (2007) L-diversity: privacy beyond k-anonymity. ACM Trans Knowl Discov Data (TKDD), vol 1
22. Mohammed N, Fung BCM, Debbabi M (2011) Anonymity meets game theory: secure data integration with malicious participants. Very Large Data Bases J (VLDBJ) 20(4):567—588
23. Mohammed N, Fung BCM, Hung PCK, Lee C-K (2010) Centralized and distributed anonymization for high-dimensional healthcare data. ACM Trans Knowl Discov Data (TKDD) 4(4):18:1–18:33
24. Narayanan A, Shmatikov V (2009) De-anonymizing social networks. In: Proceedings of the IEEE symposium on security and privacy (S&P)
25. Samarati P (2001) Protecting respondents privacy in microdata release. IEEE Trans Knowl Data Eng 13(6):1010–1027
26. Pierangela S, Latanya S (1998) Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, SRI International
27. Sweeney L (2002) k-anonymity: a model for protecting privacy. Int J Uncertain Fuzziness Knowl-based Syst 10(5):557–570
28. Wang K, Fung BCM, Yu PS (2007) Handicapping attacker's confidence. Knowl Inf Syst 11:345–368
29. Wu W, Xiao Y, Wang W, He Z, Wang Z (2010) K-symmetry model for identity anonymization in social networks. In: Proceedings of the 13th international conference on extending database technology (EDBT)

30. Wu X, Ying X, Liu K, Chen L (2009) A survey of algorithms for privacy-preservation of graphs and social networks, chapter managing and mining graph data. Kluwer Academic Publishers, The Netherlands
31. Yan X, Han J (2002) gSpan: graph-based substructure pattern mining. In: Proceedings of the 2002 IEEE international conference on data mining (ICDM), pp 721–724
32. Ying X, Wu X (2008) Randomizing social networks: a spectrum preserving approach. In: Proceedings of the 8th SIAM international conference on data mining (ICDM), pp 739–750
33. Ying X, Wu X (2009) Graph generation with prescribed feature constraints. In: Proceedings of the 9th SIAM international conference on data mining, pp 966–977
34. Zhang L, Zhang W (2009) Edge anonymity in social network graphs. In: Proceedings of the 2009 international conference on computational science and engineering, pp 1–8
35. Zheleva E, Getoor L (2007) Preserving the privacy of sensitive relationships in graph data. In: Proceedings of the 1st ACM SIGKDD international workshop on privacy, security, and trust, pp 153–171
36. Zhou B, Pei J (2008) Preserving privacy in social networks against neighborhood attacks. In: Proceedings of the 2008 IEEE 24th international conference on data engineering, pp 506–515
37. Zou L, Chen L, Tamer Özsu M (2009) K-automorphism: a general framework for privacy preserving network publication. Proc VLDB Endow 2(1):946–957

A Comprehensive Analysis of Detection of Online Paid Posters

Cheng Chen, Kui Wu, Venkatesh Srinivasan and Xudong Zhang

Abstract We initiate a systematic study to help distinguish a special group of online users, called hidden paid posters, or termed "Internet water army" in China, from the legitimate ones. On the Internet, the paid posters represent a new type of online job opportunities. They get paid for posting comments or articles on different online communities and web sites for hidden purposes, e.g., to influence the opinion of other people toward certain social events or business markets. While being an interesting strategy in business marketing, paid posters may create a significant negative effect on the online communities, since the information from paid posters is usually not trustworthy. When two competitive companies hire paid posters to post fake news or negative comments about each other, normal netizens may feel overwhelmed and find it difficult to put any trust in the information they acquire from the Internet. In this paper, we thoroughly investigate the behavioral pattern of online paid posters based on real-world trace data. We design and validate a new detection mechanism, using both nonsemantic analysis and semantic analysis, to identify potential online paid posters. Our test results with real-world datasets show a very promising performance.

Keywords Online paid posters · Behavioral patterns · Spam detection · Machine learning · Semantic analysis

C. Chen (✉) · K. Wu · V. Srinivasan
University of Victoria, 3800 Finnerty Road, Victoria, Canada
e-mail: cchenv@uvic.ca

K. Wu
e-mail: wkui@cs.uvic.ca

V. Srinivasan
e-mail: venkat@cs.uvic.ca

X. Zhang
Peking University, No. 5 Yiheyuan Road, Haidian District, Beijing, China
e-mail: zhang.xd927@gmail.com

Ö. Ulusoy et al. (eds.), *Recommendation and Search in Social Networks*,
Lecture Notes in Social Networks, DOI 10.1007/978-3-319-14379-8_6

1 Introduction

Working as an online paid poster is a rapidly growing job opportunity for many online users, mainly college students and the unemployed people. These paid posters are referred to as the "Internet water army" in China because of the large number of people who are well organized to "flood" the Internet with purposeful comments and articles. This new type of occupation originates from Internet marketing, and it has become popular with the fast expansion of the Internet. Often hired by public relationship (PR) companies, online paid posters earn money by posting comments and articles on different online communities and web sites. Companies are always interested in effective strategies to attract public attention toward their products. The idea of online paid posters is similar to word-of-mouth advertisement. If a company hires enough online users, it would be able to create hot and trending topics designed to gain popularity. Furthermore, the articles or comments from a group of paid posters are also likely to capture the attention of common users and influence their decision. In this way, online paid posters present a powerful and efficient strategy for companies. To give one example, before a new TV show is broadcasted, the host company might hire paid posters to initiate many discussions on the actors or actresses of the show. The content could be either positive or negative, since the main goal is to attract attention and trigger curiosity.

However, the consequences of using online paid posters are yet to be seriously investigated. While online paid posters can be used as an efficient business strategy in marketing, they can also act in some malicious ways. Since the laws and supervision mechanisms for Internet marketing are still not mature in many countries, it is possible to spread wrong, negative information about competitors without any penalties. For example, two competitive companies or campaigning parties might hire paid posters to post fake, negative news or information about each other. Obviously, ordinary online users may be misled, and it is painful for the web site administrators to differentiate paid posters from the legitimate ones. Hence, it is necessary to design schemes to help normal users, administrators, or even law enforcers quickly identify potential paid posters.

Despite the broad use of paid posters and the damage they have already caused, it is unfortunate that there is currently no systematic study to solve the problem. This is largely because online paid posters mostly work "underground" and no public data is available to study their behavior. We make the following contributions:

1. We collect real-world data from popular web sites regarding a famous social event, in which we believe there are potentially many hidden online paid posters.
2. We statistically analyze the behavioral patterns of potential online paid posters and identify several key features that are useful in their detection.
3. We integrate semantic analysis with the behavioral patterns of potential online paid posters to further improve the accuracy of our detection.

2 Related Work

Previous work focused on forum and blog spammers who posted advertisements or malicious URLs on the web sites. The spammers in those scenarios used software to post malicious comments on their forums and blogs to change the results of search engine or to make their sites popular. However, the definition of spam has been extended to a much wider scope. Basically, any user whose behavior interferes with normal communication or aids the spread of misleading information is specified as a spammer. Examples include comment spammers and review spammers in social media and online shopping stores.

Yin et al. [1] studied so-called online harassment, in which a user intentionally annoyed other users in a web community. They investigated the characteristics of harassment using local features, sentimental features, and contextual features. Gao et al. [2] conducted a broad analysis on spam campaigns that occurred in Facebook network. From the dataset, they noticed that the majority of malicious accounts were compromised accounts, instead of "fake" ones created for spamming. Such compromised accounts can be obtained through trading over a hidden online platform, according to [3].

Ott et al. [4] detected fictitious opinions that are deliberately and *intelligently crafted* to be authentic. To emphasize this point, the authors set strict quality control on the fictitious posts, that is, any submission found to be of insufficient quality, e.g., written for the wrong hotel, unintelligible, unreasonably short, plagiarized, etc., will be rejected. This problem is different from ours, since we do not focus on the *deceptive* opinions, but instead we aim at detecting *disruptive* comments, which are not hard to determine if a person has enough resource and time, i.e., she/he has collected a large pool of comments from different sites, a large pool of user IDs, and she/he has enough patience to read all comments and compare the comments from a same user.

The work by Jindal and Liu [5] is close to our research. They studied a dataset crawled from Amazon.com and tried to detect "opinion spam" or "review spam." In [5], the authors assumed the review spammer acts individually. In recent work [6, 7], the authors focused on detecting groups of spammers. They found that labeling groups of spammers were easier because the behavior of a group of spammers could be detected if the spammers had similar behavior when they wrote reviews for products. Our case study, however, *largely differs* from those in [5–7]. As demonstrated in this paper, paid posters involved in business conflicts have different posting patterns and do not exhibit the features presented in [5–7]. In our work, the data is not reviews for products, but any social comments, which are *shorter than the reviews* in general, regarding various aspects of the two companies, including for example the chairman, the products, and the marketing activities. As a result, the features used in our work are different from those in [5–7]. Furthermore, our semantic analysis method to improve detection performance is based on the identification of common content words and is different from those in [5–7].

3 Data Collection and Manual Labeling

3.1 Data Collection

In this paper, we analyze a business dispute between 360 and Tencent, two IT companies. [1] We collected news reports and relevant comments regarding this special social event from two famous Chinese news web sites: Sina.com [8] and Sohu.com [9]. *Sina dataset* and *Sohu dataset* will be used as the training data and test data for our detection model, respectively. We searched all the news reports and comments from Sina.com and Sohu.com over the time period from September 10, 2010 to November 21, 2010. As a result, we found 22 news reports in Sina.com and 24 news reports in Sohu.com. For each comment of each news report, we recorded the following relevant information: *Report ID*, *Sequence No.*, *Post Time*, *Post Location*, *User ID*, *Content*, and *Response Indicator* (i.e., whether the comment is a new comment or a reply to another comment).

3.2 Manual Identification

In order to analyze the behavioral pattern and classify potential paid posters and normal users, we need to find out the "ground truth" in the two datasets and we use the following guidance:

1. Users who post meaningless or contradicting comments. For example, the comments are not even slightly related to the topic in discussion. Also, a user may post multiple comments showing completely different opinions.
2. Users who post many short comments without any supporting evidence. For example, short comments like "I like 360" and "360 is good" are less likely from reasonable users involved in serious discussion.
3. Users who post negative and irrational comments to attack other persons.
4. Users who post multiple duplicate or near duplicate comments. Unlike the above three behaviors, we do not consider it as a critical criterion in labeling the datasets because both potential paid posters and normal users can have this behavior. Before making final decision, users with this behavior are carefully considered together with other criteria.

We are confident about our labels, as we believe any reasonable person would agree that a user who posts seven "I hate 360" within 2 minutes should be a potential paid poster; and any reasonable person would also agree that a user who posts both "I really like 360 because it protects my computer so well" and "It is really bad that

[1] For a full description of this dispute, please refer to http://en.wikipedia.org/wiki/360_v._Tencent.

360 steals my private information. I hate 360" should be a potential paid poster. As a result, 70 and 82 potential paid posters were identified from the Sina dataset and the Sohu dataset, respectively.

Remark 1 Finding the "gold standard" ground truth is still an open problem and no research has been able to solve this problem. Existing efforts use cross-checking among multiple annotators, as what we have done in this work. One extreme way is to hire paid posters to post fake comments and collect the corresponding texts. This method was used by Ott et al. [4], who worked on a related (but different) problem and obtained "gold standard" labels by using Amazon Mechanical Turk (AMT) to hire turkers to post fictitious hotel reviews. Nevertheless, even with such a costly method, it is difficult to obtain "gold standard" labels, because they have no guarantee that posts not from their hired tuckers are truthful. Due to the above reason, we use the word *potential* to avoid the nontechnical argument about whether a manually selected paid poster is really a paid poster. Any absolute claim is not possible unless a paid poster admits to it or his employer discloses it, both of which are unlikely to happen. The lack of "gold standard" is common in social studies, although it has been criticized and not understood by many engineers.

4 Nonsemantic Analysis

In this section, we perform statistical analysis to investigate objective features that are useful in capturing the potential paid posters' special behavior. We use Sina dataset as our training data and thus we only perform statistical analysis on this dataset. We mainly test the following four features: percentage of replies, average interval time of posts, the number of days the user remains active and the number of news reports that the user comments on. In the following figures, we use "pp" and "nu" to denote potential paid posters and normal users, respectively.

1. *Percentage of Replies*. In this feature, we calculate the probability whether a user tends to post new comments or reply to others' comments. We conjecture that potential paid posters may not have enough patience to read others' comments and reply. Therefore, they may create more new comments. Figure 1 shows the statistical results, with respect to the density and cumulative density function of reply ratio.
2. *Average Interval Time of Posts*. We calculate the average interval time between two consecutive comments from the same user. Note that it is possible for a user to take a long break (e.g., several days) before posting messages again. To alleviate the impact of long break times, for each user, we divide his/her active online time into epochs. Within each epoch, the interval time between any two consecutive comments cannot be larger than 24 h. We calculate the average interval time of posts within each epoch, and then take the average again over all the epochs. Figure 2 shows the statistical results for the probability distribution of interval posting time.

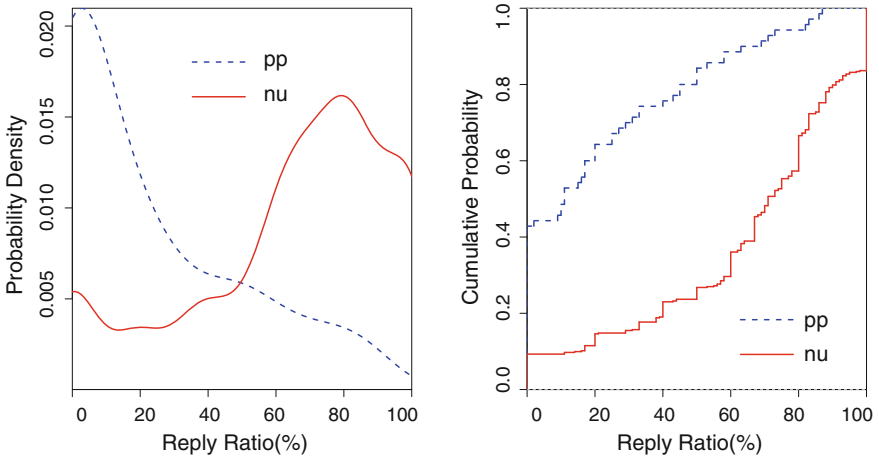

Fig. 1 The PDF and CDF of reply ratio

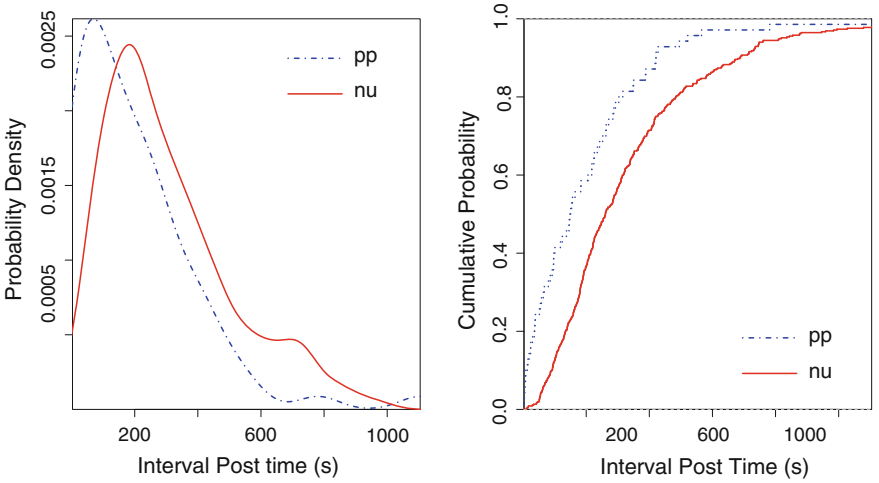

Fig. 2 The PDF and CDF of average interval time

3. *Active Days*. We analyze the number of days that a user remains active online. This information can be extracted from the time stamp of their comments. We divide the users into 7 groups based on whether they stay online for 1, 2, 3, 4, 5, 6 days and more than 6 days, respectively. Potential paid posters usually do not stay online using the same user ID for a long time. Once a mission is finished, a paid poster normally discards the user ID and never uses it again. When a new mission starts, a paid poster usually uses a different user ID, which may be newly created or assigned by the *resource team*. Figure 3 shows the statistical result. In the figures, "7" at the x-axis is the number of active days for 7 days or more.

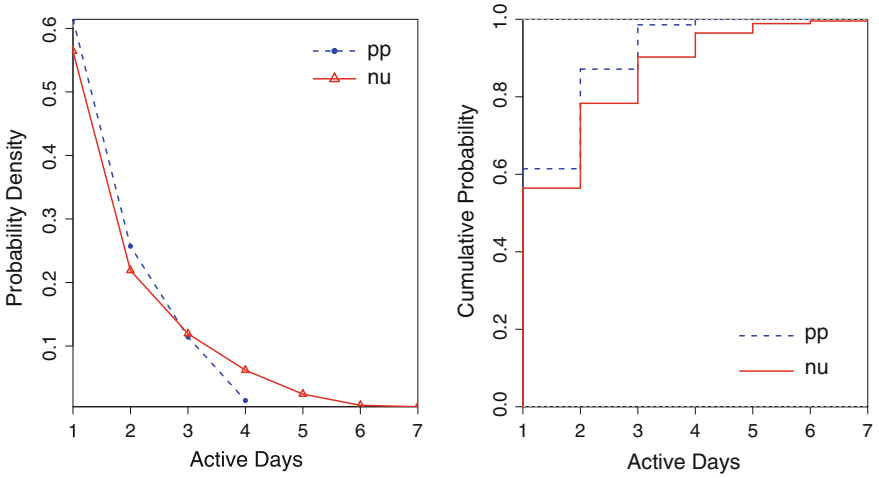

Fig. 3 The PMF and CDF of number of active days

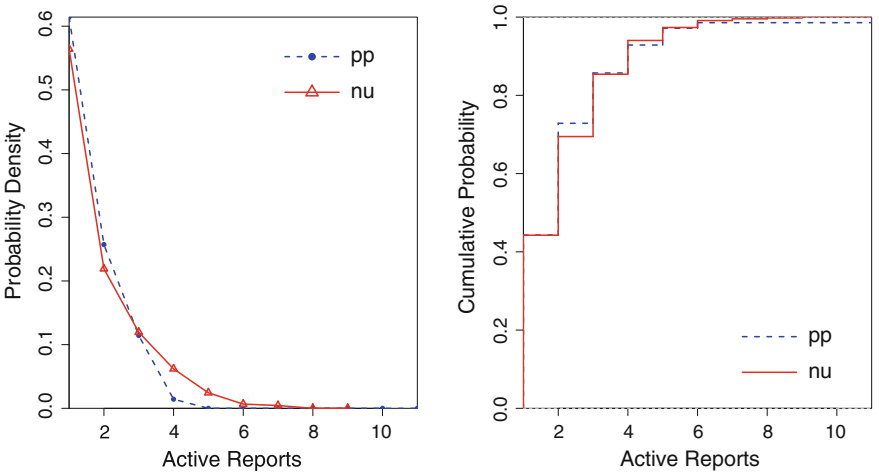

Fig. 4 The PMF and CDF of number of active news reports

4. *The Number of News Reports*. We study the number of news reports for which a user has posted comments. Both Sina and Sohu have nearly 20 news reports. Figure 4 shows the corresponding graphs.

We can derive the following conclusions from Figs. 1, 2, 3 and 4:

1. Potential paid posters tend to have smaller reply ratio.
2. Potential paid posters only care about finishing their jobs as soon as possible and do not have enough interest to get involved in the online discussion. 60 % potential paid posters post comments within interval time of 200 seconds.

3. Potential paid posters are not willing to stay for a long time. They instead tend to accomplish their assignments quickly and once it is done, they would not visit the same web site again.
4. Potential paid posters and normal users have similar distribution with respect to the number of commented news reports. This indicates that the number of commented news reports alone may not be a good feature for the detection of potential paid posters.

5 Semantic Analysis

An important criterion in our manual identification of a potential paid poster is to read his/her comments and make a choice based on common sense and online experience. While it is hard to design a detection system that understands the meaning of a comment, we observed that potential paid posters tend to post similar comments on the web. In many cases, a potential paid poster may copy and paste existing comments with slight changes. This provides the intuition for our semantic analysis technique.

Our basic idea is to search for similarity between comments. To do this, we first need to overcome the special difficulty in splitting a Chinese sentence into words and phrases. We used a famous Chinese splitting software, called ICTCLAS2011 [10], to cut a sentence into words. It translates a sentence into a list of content words. For a given pair of comments, we compare the two lists of content words. As mentioned before, a paid poster may make slight changes before posting two similar comments. Therefore, we may not be able to find an exact match between the two lists. We first find their common content words, and if the ratio of the number of common content words over the length of the shorter content word list is above a threshold value (e.g., 80 % in our later test), we conclude that the two comments are similar. If a user has multiple pairs of similar comments, the user is considered a potential paid poster. Note that similarity of comments is not transitive in our method.

We found that a normal user might occasionally have two *identical* comments. This may be caused by the slow Internet access, due to which the user presses the *submit* button twice before his/her post is displayed. Our manual check of these users confirmed that they are normal users, based on the content they posted. To reduce the impact of the "unusual behavior of normal users," we set the threshold of similar pairs of comments to 3. This threshold value is demonstrated to be effective in addressing the above problem.

We performed the semantic analysis over the Sina dataset. The result is shown in Fig. 5.

In the figure, "6" on the x-axis means the number of similar pairs is larger than or equal to 6. The two groups of users obviously show different patterns. Normal users have much higher probability to post different comments. In the opposite, the potential paid posters have many similar pairs of comments in their profiles. Therefore, it is important to monitor the number of similar pairs of comments in a user's profile as it is a significant indication of malicious behavior.

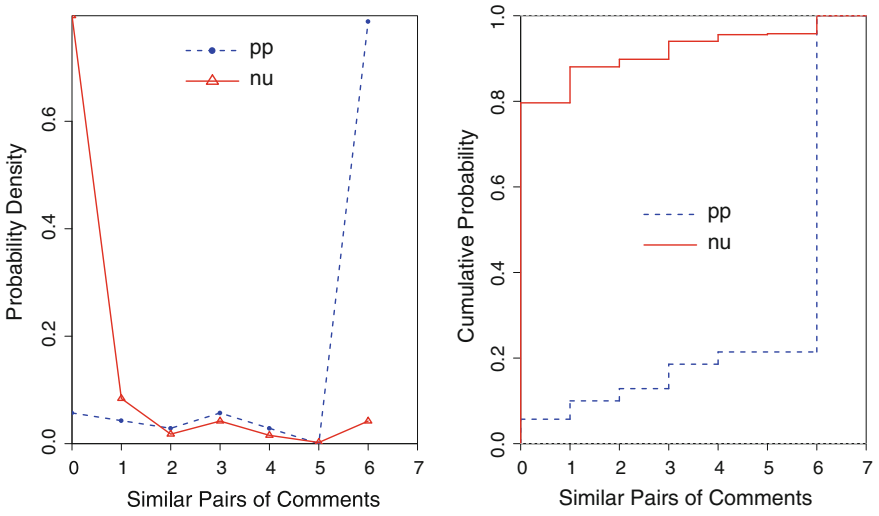

Fig. 5 The PMF and CDF of the number of similar pairs of comments

6 Classification

The objective of our classification system is to classify each user as a potential paid poster or a normal user using the features investigated in Sects. 4 and 5. According to the statistical and semantic analysis results, we found that any single feature is not sufficient to locate potential paid posters. Therefore, we compare the performance of different combinations of the five features discussed in the previous two sections in our classification system. We model the detection of potential paid posters as a binary classification problem and solve the problem using a support vector machine (SVM) [11].

We used LIBSVM [12] as the tool for training and testing. By default, LIBSVM adopts a radial basis function [11] and a 10-fold cross-validation method to train the data and obtain a classifier. We did not tune any model parameter of libsvm and liblinear. All results came from the default settings, so that we could compare the results in a general way. The Sina dataset is divided into 10 subsets of equal size. Then the model is trained on the 9 subsets and tested on the remaining subset. The process returns a model with the highest cross-validation accuracy. After training the classifier with the Sina dataset, we used the classifier to test the Sohu dataset.

We evaluate the performance of the classifier using the four metrics: precision, recall, f measure and accuracy, defined as follows:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{1}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{2}$$

$$\text{F measure} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

$$\text{Accuracy} = \frac{\text{True Negative} + \text{True Positive}}{\text{Total Number of Users}} \tag{4}$$

6.1 Classification Without Semantic Analysis

To simplify the notation, the five features, *reply ratio*, *average interval posing time*, *active days*, *active reports*, and *degree of similarity* are labeled as features "1", "2", "3", "4" and "5," respectively. The first four features are statistical ones while the last is a semantic feature.

We firstly focus on the classification only using statistical analysis results based on the four statistical features. Different combinations are applied to test their performance for identification. We train the SVM model using the Sina dataset with different combinations of the features. Then we test the model with the Sohu dataset to see the performance. Note that combinations that result in 0 true positive and 0 false positive are not considered. The results are shown in Fig. 6.

Although the (1-2)-feature test has the highest precision, its recall and f measure are very low, showing that the (1-2)-feature can hardly separate different classes of users. This result suggests that the first two features lead to significant bias and we need to add more features to our classifier. With features 3 and 4 considered, we observe better performance. For example, the (1-2-3)-feature test has better performance over all the metrics, except precision.

Nevertheless, we notice when we use only nonsemantic features to train the SVM model, the overall performance on the four metrics is not good enough to claim acceptable performance. Particularly, the low precision and accuracy results indicate that the SVM classifier using the four nonsemantic features as its vector set is unreliable and needs to be improved further. We achieve this by adding the semantic analysis to our classifier.

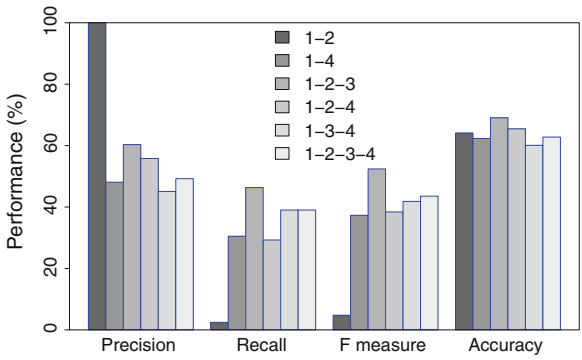

Fig. 6 The performance of different combinations of statistical features

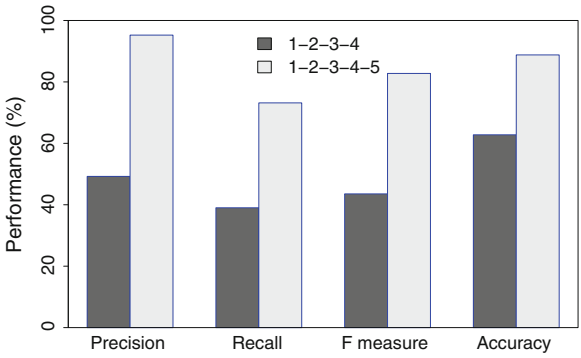

Fig. 7 The performance of statistical and semantic features

6.2 Classification with Semantic Analysis

As described in Sect. 5, we have observed that online paid posters tend to post a larger number of similar comments on the web. Based on this observation we have designed a simple method for semantic analysis. We test the performance of all the five features. After integrating this semantic analysis method into our SVM model, we observed the much improved performance results as shown in Fig. 7.

The results clearly demonstrate the benefit of using semantic analysis in the detection of online paid posters. The precision, recall, f measure, and accuracy have been improved to 95.24, 73.17, 82.76 and 88.79 %, respectively. Based on these results, the semantic feature can be considered as a useful and important supplement to other features. The reason why the semantic analysis improves performance is that online paid posters often try to post many comments with some minor changes on each post, leading to similar sentences. This helps the paid posters post many comments and complete their assignments quickly, but also renders it easy to detect them.

Having shown that the five proposed features together lead to higher performance, we add additional tests with random forest [13]. Random forest consists of multiple decision trees whose prediction is easier to explain (i.e., the criteria for making predictions) and no parameter tuning is required. In the learned tree structures, branches are conjunctions of features that lead to class labels which are represented by leaves. In practice, it will aid better interpretation of decision-making. The performance of SVM and random forest is shown in Fig. 8. The result shows that random forest does not perform as well as SVM.

6.3 Classification Using only Text Information

As a comparison to the previous method, we use a typical information retrieval approach to identify potential paid posters in this subsection. We now use only text information (individual words in comments) for training the classifier. Specifically, we treat each user's comments as an individual document and it becomes a binary

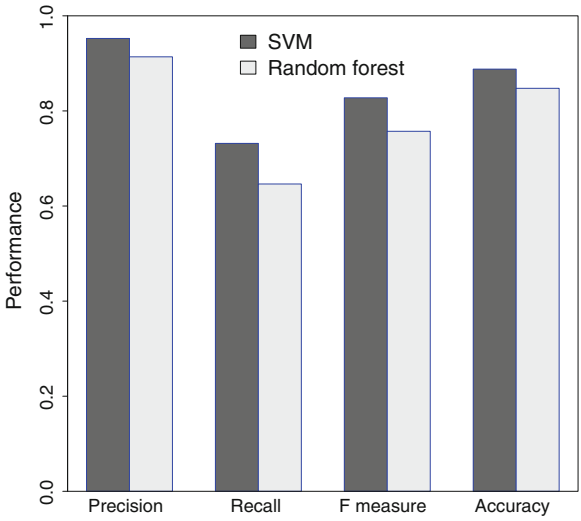

Fig. 8 Performance comparison of SVM and random forest

document classification problem; to detect potential paid posters is to classify each document into two distinct groups (i.e., malicious and normal).

6.3.1 Feature Selection

We use the Chi-square method [14, 15] to retrieve a bag of feature words, a standard methodology of extracting features in documentation classification.

We define variables A, B, C, and D in Table 1. For example, A is the number of paid posters who have a specific word in the comments. D is the number of normal users who do not have the specific word.

After we collect the statistic information for every individual word, we can then compute Chi-square values. The Chi-square value of a word in the document collection is defined as

$$\text{chisquare}(\text{word}, \text{classification}) = \frac{(AD - BC)^2}{(A + B)(C + D)} \tag{5}$$

We compute the Chi-square value for each word in the training document collection, sort them in descending order and retrieve the first d words as the bag of the most predictive features.

Table 1 Chi-square feature selection

Feature selection	Paid posters	Normal users	Total
Has word	A	B	$A + B$
Not word	C	D	$C + D$
Total	$A + C$	$B + D$	N

6.3.2 Vectorization

After selecting feature words from the document collection, we can then vectorize each document by associating it with a vector of dimension d. We compute the weight for each dimension using the TF/IDF approach [16].

6.3.3 Classifier

To study the performance of text information for this document classification problem, we explore different nonlinear classifiers as well as linear ones on our dataset and compare their prediction results. In the following, we use Liblinear [17] for the linear classifier.

Compared to the general-purpose SVM solver Libsvm, Liblinear is exclusively used for linear classification, i.e., it supports logistic regression and linear support vector machines. Without using kernels, Liblinear can train a much larger set via a linear classifier. Consequently, Liblinear is considered a better choice over Libsvm when handling large-scale datasets (e.g., document classification) for which using nonlinear mappings does not provide additional benefit.

Tables 2 and 3 list candidate models to be tested in the experiment.

6.3.4 Performance Evaluation

In order to show the impact of different dimensions, we use four settings for the following tests, i.e., $d = 100$, $d = 200$, $d = 300$, and $d = 400$. The results are shown in Fig. 9.

Table 2 Libsvm kernel types

Kernel type	Description
t0	Linear: $u' * v$
t1	Polynomial: $(\gamma * u' * v + \text{coef0})^{\circ}$
t2	Radial basis function (RBF): $\exp(-\gamma * \lvert u - v\rvert^2)$
t3	Sigmoid: $\tanh(\gamma * u' * v + \text{coef0})$

Table 3 Liblinear solver types

Solver type	Description
s0	L2-regularized logistic regression (primal)
s1	L2-regularized L2-loss support vector classification (dual)
s2	L2-regularized L2-loss support vector classification (primal)
s6	L1-regularized logistic regression

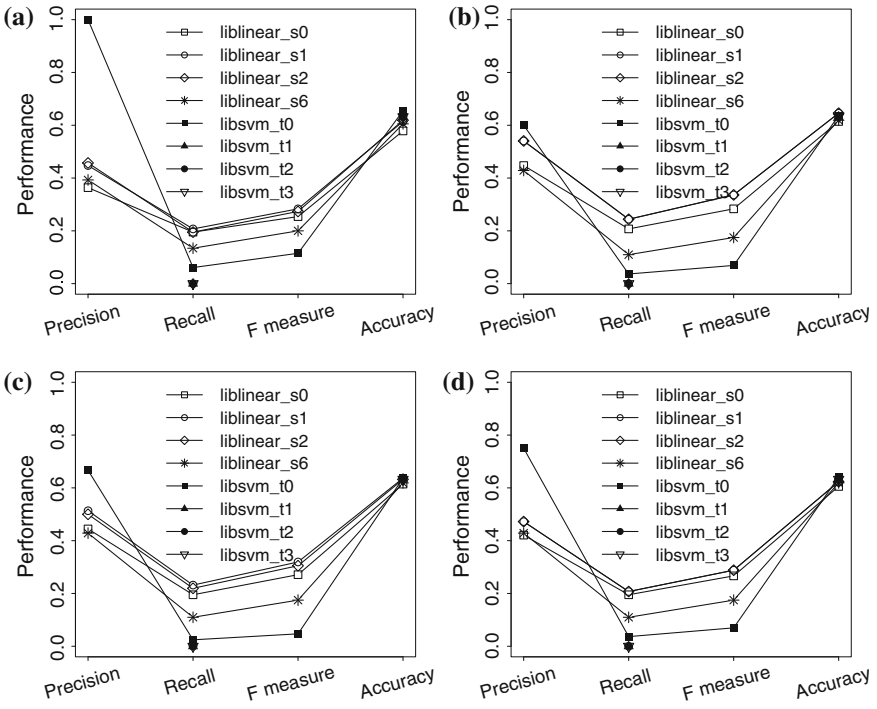

Fig. 9 Performance curves for different dimensions for Libsvm and Liblinear **a** $d = 100$ **b** $d = 200$ **c** $d = 300$ **d** $d = 400$

From Fig. 9 we observe that models trained by Liblinear have better performance over the ones trained by Libsvm. Specifically, Liblinear models of $d = 200$ have the overall best performance. Recall and f measure of Liblinear are significantly higher than Libsvm, even if precision of Libsvm with t0 is the highest. Note that metrics of precision and f measure for Libsvm models with nonlinear kernels (t1, t2, and t3) in the figures are not available (corresponding to the missing points in Fig. 9), because those models only return negative predictions. It indicates that nonlinear SVM classifiers are not valid in this high-dimensional classification problem. All valid models have similar accuracy measurement.

In addition, an interesting observation is that the best Liblinear model does not exceed the performance of Libsvm model trained by (1-2-3-4)-feature, which is described in previous subsections. The reason is that the high-dimensional feature space is too sparse to facilitate the learning algorithm. The sparsity is due to the fact that a user's comments tend to be short and the selected feature words cannot provide enough coverage even if we group each user's comments.

In order to evaluate the performance over all features mentioned in this paper, we add 200-dimension text information into the feature space, labeled by (1-2-3-4-5). We then use Liblinear to train a linear classifier and evaluate it over the Sohu test set. The results are shown in Figs. 10 and 11.

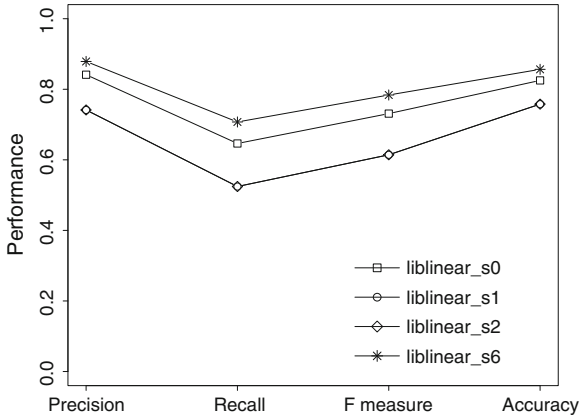

Fig. 10 Liblinear combination of features 1-2-3-4-5 and 200-dimension text feature

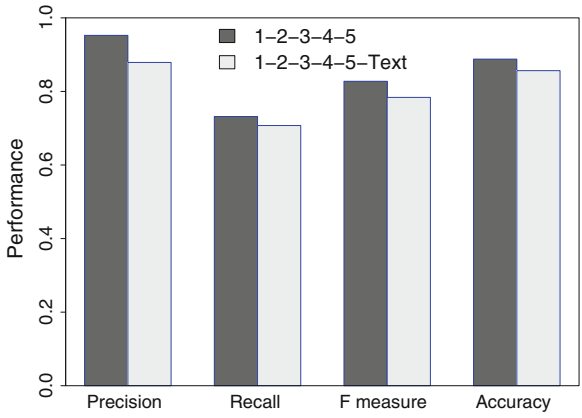

Fig. 11 Performance comparison with/without 200-dimension text feature

Figure 10 shows that Liblinear with s6 outperforms other Liblinear models. In Fig. 11, we compare the best results of Liblinear with s6 (including 200-dimension text feature) to the previous one (Libsvm performance excluding text feature). It shows that adding 200-dimension text feature would unfortunately harm the overall performance.

6.4 Classification Using Unsupervised Learning

For unsupervised learning, we firstly merged Sina dataset and Sohu dataset and applied K-means clustering algorithm to obtain K clusters. If the five features have the ability to distinguish paid posters from normal users, we expect that paid posters should be grouped into a cluster. In our work, we only need two clusters, one for paid posters and one for normal users. Furthermore, to check the reliability of our features, we studied two more cases, corresponding to $K = 3$ and $K = 4$.

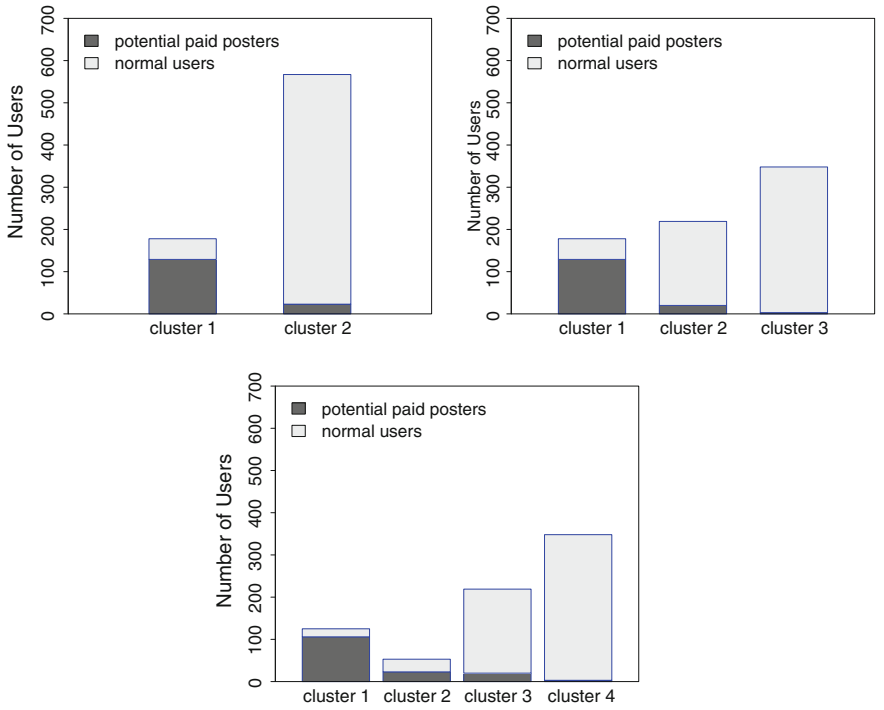

Fig. 12 Clustering analysis $K = 2, 3, 4$

Figure 12 shows the size of each cluster as well as the number of potential paid posters and normal users in each cluster.

From the figures, we notice that when $K = 2$, a large proportion (approximately 85 %) of potential paid posters is assigned to a particular cluster (cluster 1). When $K = 3$ and $K = 4$, cluster 1 (the group of paid posters) remains stable. Nevertheless, the other cluster (the group of normal users) is further divided into smaller clusters. This phenomenon suggests that although normal users might have different behavioral patterns, they in general behave much different from potential paid posters.

We also notice that a small number of normal users are assigned to cluster 1. This is because our manual labeling uses human intelligence (refer to Sect. 3), which cannot be completely captured by the five features. This poses the challenge of developing more intelligent detection mechanism for our future work.

We now compare the clustering model with the supervised model. We train a Gaussian mixture model [18], a generalization of K-means, on Sina dataset and test it on Sohu dataset. It incorporates information about the means ($\boldsymbol{\mu}$) and covariance matrix ($\boldsymbol{\Sigma}$) of features of the data. As an unsupervised approach, it applies expectation–maximization algorithm [19] to estimate the model parameters. Its prediction is based on the sample's probability of being assigned to each cluster. We compare its performance with that of SVM in Fig. 13.

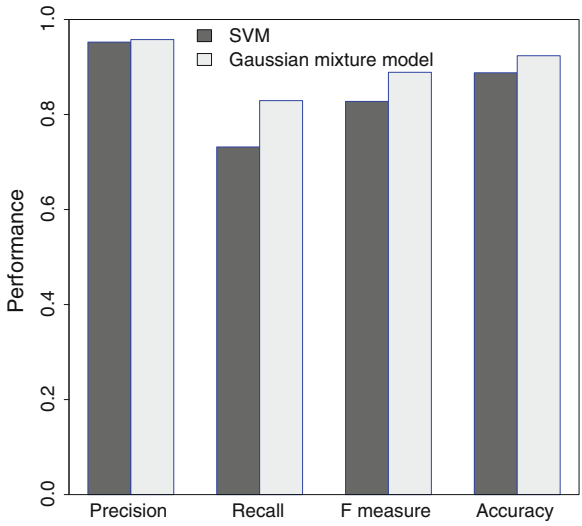

Fig. 13 Performance comparison of SVM and Gaussian mixture model

In Fig. 13, we can see that Gaussian mixture model outperforms SVM in all metrics, even the lowest recall measure exceeds 80 %. The result implies that not all features of the data satisfy the assumption of independent and identical distribution. In addition, the test using Gaussian mixture model also demonstrates the effectiveness of the five proposed features to differentiate the malicious from the normal.

7 Conclusions and Future Work

Detection of paid posters behind social events is an interesting research topic and deserves further investigation. In this paper, we disclose the organizational structure of paid posters. We also collect real-world datasets that include abundant information about paid posters. We identify their special features and develop effective techniques to detect them. The performance of our classifier, with integrated semantic analysis, is quite promising on the real-world case study, as confirmed in both supervised learning and unsupervised learning techniques.

This work is our preliminary effort to battle online paid posters. It requires a prolonged and systematic effort to reach a complete solution, as the online paid posters evolve continuously and present new challenges to the detection mechanism. We will further improve our detection system and evaluate the system in a broader and larger dataset. We wish this work would attract further research activities.

Acknowledgments We thank Natural Sciences and Engineering Research Council of Canada (NSERC) and Mathematics of Information Technology And Complex Systems (MITACS) for the funding support. We thank MIT Tech. Review [20] for announcing our work to the public.

References

1. Yin D, Xue Z, Hong L, Davison B, Kontostathis A, Edwards L (2009) Detection of harassment on web 2.0. In: Proceedings of the content analysis in the web 2
2. Gao H, Hu J, Wilson C, Li Z, Chen Y, Zhao BY (2010) Detecting and characterizing social spam campaigns. In: ACM conference on computer and communications security, pp 681–683
3. Staff E (2010) Verisign: 1.5 m facebook accounts for sale in web forum. http://www.pcmag.com/article2/0,2817,2363004,00.asp
4. Ott M, Choi Y, Cardie C, Hancock JT (2011) Finding deceptive opinion spam by any stretch of the imagination. In: ACL, pp 309–319
5. Jindal N, Liu B (2008) Opinion spam and analysis. In: Proceedings of the international conference on web search and web data mining, WSDM'08, ACM, New York, pp 219–230
6. Mukherjee A, Liu B, Glance NS (2012) Spotting fake reviewer groups in consumer reviews. In: Proceedings of the 21st international conference on World Wide Web: pp 191–200
7. Mukherjee A, Liu B, Wang J, Glance NS, Jindal N (2011) Spotting fake reviewer groups in consumer reviews. In: WWW: pp 191–200
8. Sina.com. www.sina.com.cn Accessed Jan 2011
9. Sohu.com. www.sohu.com Accessed Jan 2011
10. ICTCLAS2011. http://hi.baidu.com/drkevinzhang/home Accessed Mar 2011
11. Cristianini N, Shawe-Taylor J (2006) An introduction to support Vector Machines and other kernel-based learning methods. Cambridge University Press, Cambridge
12. Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. ACM transactions on intelligent systems and technology, vol **2**(27), pp 27:1–27:27 Software available at http://www.csie.ntu.edu.tw/.
13. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
14. Zheng Z, Wu X, Srihari R (2004) Feature selection for text categorization on imbalanced data. SIGKDD Explor Newsl 6(1):80–89
15. Forman G (2003) An extensive empirical study of feature selection metrics for text classification. J Mach Learn Res 3:1289–1305
16. Joachims T. (1997) A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In: Proceedings of the fourteenth international conference on machine learning ICML'97, Morgan Kaufmann, San Francisco, pp 143–151
17. Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) Liblinear: a library for large linear classification. J Mach Learn Res 9:1871–1874
18. Marin JM, Mengersen K, Robert CP (2005) Bayesian modelling and inference on mixtures of distributions. Handb Stat 25:459–507
19. Dempster AP, Laird NM, Rubin DB et al (1977) Maximum likelihood from incomplete data via the em algorithm. J R Stat Soc 39(1):1–38
20. MIT-Tech-Review. www.technologyreview.com/blog/arxiv/27357/ Accessed Nov 2011

An Improved Collaborative Recommendation System by Integration of Social Tagging Data

Sogol Naseri, Arash Bahrehmand and Chen Ding

Abstract Recently a lot of research efforts have been spent on building recommender systems by utilizing the abundant online social network data. In this study, we intend to enhance the recommendation accuracy via integrating social networking information with the traditional recommendation algorithms. To achieve this goal, we first propose a new user similarity metric that not only considers tagging activities of users, but also incorporates their social relationships such as friendship and membership, in measuring the closeness of two users. Then we define a new item prediction method which makes use of both user-to-user similarity and item-to-item similarity. Experimental outcomes on Last.fm data produce the positive results that show the accuracy of our proposed approach.

Keywords Social networking · Collaborative filtering · Social tagging · User similarity

1 Introduction

Due to the increasing reputation of social networking web sites, researchers have drawn attention toward refining these growing data to alleviate undesirable effect of information overload. Information overload is a predominant and growing issue that designates to an incongruity between the amount of information, and the

C. Ding (✉) · S. Naseri
Department of Computer Science, Ryerson University, Toronto, Canada
e-mail: cding@ryerson.ca

S. Naseri
e-mail: sogol.naseri@ryerson.ca

A. Bahrehmand
Department of Information and Communications Technologies,
Universitat Pompeu Fabra Barcelona, Barcelona, Spain
e-mail: Arash.bahrehmand@upf.edu

Ö. Ulusoy et al. (eds.), *Recommendation and Search in Social Networks*,
Lecture Notes in Social Networks, DOI 10.1007/978-3-319-14379-8_7

ability to analyze that information [1]. One of the most significant attempts to make recommendations more precise is ranking information relevancy based on the assumption that people filter information collaboratively or socially and are seeking information according to what others have already discovered and appraised.

The social tagging system is a rich environment, which enables researchers to analyze the user taste and the items' attributes from the social relations (friendship, membership, etc.) between the users. Although social tagging systems have emerged to overcome the information overload problem, the simplicity and ease of use of tagging, in some cases, not only increases the amount of available information, but also gives rise to some unfamiliar complications. To the best of our knowledge, most of the current tag-based systems did not distinguish the difference of shared tags on common items from shared tags that are assigned to different items, which is a key factor in measuring the similarity of two users. In this study, we measure the similarity of two users' opinions about a particular item by calculating their shared tags on that item.

One of the most prevalent advantages of using social networking information is taking advantage of friendship information in which users define their neighbors explicitly. This paper attempts to distinguish those friends that the target user relies more on them in terms of accepting their recommendations. The other widespread social networking activity of users is participating in groups. As a part of our system, we come up with measuring the level of the participation of users in the same groups with the intention of discriminating users based on their activities in groups. In this sense, the more two users have the same experience in a group, the higher similarity value can be considered for them.

In our approach nearest neighbors are calculated by a new similarity metric based on their tagging histories and social activities, whereas most of the traditional collaborative filtering approaches obtain top K nearest neighbors for a user based on her/his rating, using Pearson correlation or cosine correlation formulas [2, 3]. The proposed similarity metric includes the combination of three main similarity metrics: similarity based on common tags on common items, similarity based on friendship and similarity based on membership. Moreover, we introduce a new recommendation method that applies the item similarity as well as user similarity for suggesting items. Finally, in order to evaluate how successful our proposed approach is in predicting user interests, we implement our algorithm on Last.fm dataset. Last.fm[1] is a music recommender web site that provides users with the facilities of tagging resources, making friendship and joining groups.

The rest of this paper is organized as follows: In Sect. 2, the literature review is presented. In Sect. 3, our proposed approach is discussed. In Sect. 4, the experimental results and analyses are given. In Sect. 5 the presented work is summarized and our future work is discussed.

[1] http://www.last.fm/.

2 Related Work

With the dramatic expansion of the internet, we have witnessed the emergence of huge amount of unstructured data, which makes the process of finding appropriate information a challenging task for end users. Recommender systems are supposed to reduce information overload through personalized recommendations based on user preference and behavior. A number of studies examine various aspects of recommender systems in different personalized services: (i) Collaborative filtering, (ii) Social tagging systems, and (iii) Collaborative social tagging systems.

2.1 Collaborative Filtering

Collaborative filtering (CF) is one of the most popular recommendation techniques, which is used to filter information [4, 5]. The main idea of this kind of methods is that if users shared the same interests in the past—if they viewed same movies, for instance—they will also have similar behavior in their future decisions [6–8]. CF algorithm mainly considers the rating of users which could be explicitly assigned from 1 to 5, or it could be identified implicitly if the user bought or selected that item [9].

CF algorithms can be categorized into three main groups: Memory-based CF, Model-based CF, and Hybrid CF. In order to find K nearest neighbors in memory-based CF the similarity between two users or items can be calculated based on Pearson Correlation Coefficient similarity, cosine similarity, or Jaccard similarity [4, 10]. Then, according to the nearest neighbors' opinions the most interesting items for the target user are recommended [3, 10, 11].

2.2 Social Tagging Systems

Tags are some keywords assigned to an object (photos, music tracks, videos, etc.) to provide a meaningful description for it [12]. Tagging techniques in the tag-based systems such as: Flicker,[2] Del.icio.us,[3] Last.fm and CiteU-Like,[4] provide a rich method for organizing user contents, managing and locating relevant information. For example, users of Flicker harness tags to manage their photos and to explore other interesting photos. On Del.icio.us, tags are used to help users organize, share, and discover bookmarks. In some popular recommendation websites such as Del.icio.us and last.fm, tag and social networking information are associated. In Last.fm, people

[2] http://www.flickr.com/.

[3] https://delicious.com/.

[4] http://www.citeulike.org/.

assign tags to the tracks, albums or artists and they can make friends or join into their interesting groups. Also, CiteU-Like is a free service to store, share, and organize academic papers.

Usually, the assigned tags help users revisit their previously selected resources or searching for favorite items of other users. With the purpose of measuring the usefulness of tags in generating personalized recommendations, some researchers extract the semantic meaning of tags to find tag similarity [13–15]. Also, in [16–18], there are several proposed methods which use tags to compute the similarity of users. Authors in [12] present a tag-based recommender system which recommends web pages based on their tag similarity. In other words, for the purpose of suggesting personalized resources, an extension method is proposed for computing similarity between tags; in a way that similarity calculation is a combination of cosine similarity metric with other factors such as tag frequency, tag popularity, and affinity between user and tag. In [14] a new method is proposed which incorporates tags in CF algorithm and applies three two-dimensional correlations for items, tags, and users. Tags not only are used to organize contents and define a clue that why the user liked something, but also are beneficial for users to help them find their interesting items [15]. In addition, there are some tag-based approaches that exploit semantic web strategies for extracting the knowledge behind system resources, however, are not covered in this study.

2.3 Collaborative Social Tagging System

In the past few years, application of extracted social data from social web sites has become increasingly popular; in a way, fusing social networking information with recommender systems for increasing the level of personalization has received a significant attention from the research communities [19–21]. Tag-based systems enriched our description of items while social networking systems opened up the door for developing more accurate graphs arising from users' relationships. Moreover, according to the sparsity problem that item-based or user-based traditional CF algorithms suffer from, researchers attempted to combine other sources of data with traditional filtering methods. Therefore, some researchers came up with the idea of using trust theory. Authors in [22–25] consider trust relations of users to improve the accuracy of item recommendation. The trust theory believes that people prefer recommendations from the people they know or trust. In doing so, a trust value is obtained from users when they define how much they trust the people that they know.

By comparing recommendations from friends with generated recommendations via collaborative methods, it could be inferred that friends' recommendations are preferred. It means friends usually share common tastes and interests and it is easy to find the trusted users by the given user based on her/his friendship relations. The presented method in [26] enhances the accuracy of recommender system when the social networking information is incorporated to CF algorithm. In [26], data of users' preference ratings and their social network relations are collected. Then,

the nearest neighbors are recognized by Pearson correlation coefficient similarity metric. Finally, if the social network members are in the list of nearest neighbors the member's preference is amplified.

Among proposed methods for item recommendations, some of them have been evaluated based on Last.fm data sets. Authors in [27] used last.fm as an appropriate environment for testing their probabilistic generative model, called social influenced selection model (SIS), which incorporates user behavior, social influence, and item content in measuring item similarity. In [28] Last.fm dataset is adopted to explore the role of friendship information in track recommendation. This study came up with a considerable effectiveness in predicting future musical preferences of users based on social information.

Although some researchers apply friendship information, others attempted to fuse membership for item recommendation [29, 30] based on the belief that joining group is a direct indicator of the user's interest comparing to friendship since making friend can be done for various reasons. For example, in [31] membership information is used in Orkut[5] social network site in order to recommend communities to members. This approach presents a new collaborative filtering that takes advantage of overlapping membership of pairs in communities. In fact, all the members of a given community get the same recommendation when they visit their community's page.

Some researchers believed that combining other data sources with friendship information instead of purely concentrating on friendship can improve the accuracy of recommender systems [32]. Therefore, in [32] both friendship and membership information are used while being combined with traditional CF to predict items more precisely. Moreover, in order to explore the effect of both membership and friendship information, two methods—random walk graph with CF and weighted neighborhood similarity, are presented. The proposed study compares these two methods while those two kinds of social network information are fused on random walk graph with CF and neighborhood similarity. In tag-based collaborative filtering recommender systems such as [33, 34], first, a tag weight calculation is computed for a user or item, and then based on the calculated tag weight a probability score is calculated to predict items that target user is interested in.

To sum up, in this paper, we developed a new approach to find nearest neighbors based on combination of CF and social tagging relations to enhance recommendation accuracy. Compared with other papers, when incorporating social data, our first contribution is recognizing the most effective friends instead of just considering all friends equally, and the second one is recognizing members who contribute more to the community instead of treating all members equally. When considering tags, our approach focuses on common tags on common items to improve recommendation accuracy. Calculation in [26] is based on user ratings. In this study, we do not have direct rating information available, but we build a rating matrix for users and items in a way that if a user assigns a tag to an item the rating matrix is 1, otherwise it is 0.

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

3 Social Tagging-Based Collaborative Recommendation

In this section, we describe the two main stages of our collaborative filtering methods: measuring similarity and making recommendations. The former demonstrates a new similarity metric that combines explicit and implicit relationships between users, while the latter is an item recommendation method based on a simple weighted average approach.

3.1 Measuring Similarity

Our proposed similarity metric contributes three main innovations in addressing user similarity based on item-tag pair, friendship, and membership.

3.1.1 Item-Tag-based Similarity

Tag-based systems can express user preferences for a resource by providing specific ways for web users to express their personal opinions in their own words. Measuring the user similarity only based on common tags is not appropriate, since the possibility of having two users with many common tags while most of those tags are not assigned to any shared items, remains open. On the other hand, measuring the similarity of users only based on common items could also not be a precise measure. Because maybe these users have many common items; however, they assigned different tags to those items. In our item-tag similarity metric, we only focus on common tags, which are assigned to same items. We define the similarity value between two users u and v as below:

$$TSim_{u,v} = \frac{\sum_{i \in (I_u \cap I_v)} \left(\frac{(|Tuv_i|)^2}{|Tu_i| * |Tv_i|} \right)}{\text{Max}(|I_u|, |I_v|)} \tag{1}$$

where T_{ui} is the set of tags that user u assigned to item i, T_{uvi} is the set of shared tags between users u and v on item i and Iu is the item set of user u. Moreover, $\max(|Iu|, |Iv|)$ indicates the maximum number of items selected by u and v. In (1), $\frac{(|Tuv_i|)^2}{|Tu_i| * |Tv_i|}$ measures how similar is the opinion of u and v for a shared item i. Therefore, the more overlap between their tag sets on item i, the more similar of their judgments of item i.

In order to determine an appropriate denominator we analyze four candidate operations for the denominator:

- The minimum number of items selected by two users
- The number of items selected by one of the users
- The number of common items selected by two users
- The maximum number of items selected by two users.

Suppose that we have three users who have assigned tags to items as shown in Table 1.

Table 1 Sample user matrix

	i_1	i_2	i_3	i_4
u_1	t_1, t_3	t_2, t_4		
u_2	t_1, t_3	t_2, t_4		
u_3	t_1, t_3	t_2, t_4	t_5	t_6

In the following, we measure the similarity of u_1 with u_2 and u_3 based on the four candidates of the denominator.

- Min($|Iu|, |Iv|$): If we divide the numerator by the minimum value then we have:

$$TSim_{u1,u2} = \frac{\frac{2}{2} * \frac{2}{2} + \frac{2}{2} * \frac{2}{2}}{\text{Min}(2, 2)} = 1$$
$$TSim_{u1,u3} = \frac{\frac{2}{2} * \frac{2}{2} + \frac{2}{2} * \frac{2}{2}}{\text{Min}(2, 3)} = 1$$

  It means that the similarity of u_1 and u_2 is equal to the similarity of u_1 and u_3. However, it is quite obvious that the former similarity should be higher than the latter because two users u_1 and u_2 assign the exact same tags to two items.
- $|Iu|$ or $|Iv|$: If denominator is the number of items that user u selected or the number of items that user v selected, $Tsim_{u,v}$ is not equal to $Tsim_{u,v}$ and the similarity is not symmetric.
- ($|Iu \cap Iv|$): If the denominator is the number of common items between the two users then a significant problem may occur. If user u and user v do not share any common items, the denominator is 0.
- Max($|Iu|, |Iv|$): In this case the similarity values are as follows, which are more reasonable values:

$$TSim_{u1,u2} = \frac{\frac{2}{2} * \frac{2}{2} + \frac{2}{2} * \frac{2}{2}}{\text{Max}(2, 2)} = 1$$
$$TSim_{u1,u3} = \frac{\frac{2}{2} * \frac{2}{2} + \frac{2}{2} * \frac{2}{2}}{\text{Max}(2, 3)} = \frac{2}{3}$$

This example explained our conclusion to take the maximum number as the best option for the denominator.

3.1.2 Friendship-Based Similarity

Currently, the impact of friend's interests on recommendations has not been fully explored. For instance, the recommendation system should be able to account for varying tastes among friends, and evaluate whether the overlap in those tastes will result in a successful recommendation. Therefore, a framework can be developed for assessing how much interests affect friendships in parallel with how friendships affect interests. As pointed out before, traditional collaborative filtering approaches obtain top K nearest neighbors for a target user; however, the neighbors' order can

be changed if friendships are taken into account. In order to change the priority of neighbors, a reinforcement method is applied based on the level of friendship. Due to the fact that two users may be friends in a social tagging system but they may not share any common interest on most of the items, we recognize those friends who have been trusted mostly by the target user and shared the similar interests with her/him.

Firstly, for a user u in the system we calculate $AvgFu$ (the average of the item-tag-based similarities of all the friends of u) and then we amplify the item-tag-based similarity of u and v if user v is among those friends of u who are mostly similar.

In Eq. (2), $AvgFu$ is the average of the item-tag similarities of all friends of u. Since the value of $AvgFu$ may be different than that of $AvgFv$, the similarity based on friendship is an asymmetric similarity. In this sense, we came up with a simple algorithm that is formally given as follows:

$$FSim_{u,v} = TSim_{u,v}^{\frac{1}{1+(TSim_{u,v}-AvgF_u)}} \tag{2}$$

Input: user u and v
Step1: Calculate $AvgF_u$ and $AvgF_v$
Step2: **IF** v is friend of u

a. IF $TSim_{u,v}>AvgF_u$, **THEN** $FSim_{u,v} = TSim_{u,v}^{\frac{1}{1+(TSim_{u,v}-AvgF_u)}}$;

b. IF $TSim_{u,v}>AvgF_v$, **THEN** $FSim_{v,u} = TSim_{u,v}^{\frac{1}{1+(TSim_{u,v}-AvgF_u)}}$;

ELSE

c. $FSim_{u,v}$=0;

d. $FSim_{v,u}$=0;

Output: $FSim_{u,v} \&\& FSim_{v,u}$

3.1.3 Membership-Based Similarity

The membership information reflects the behavior of a user in her/his shared community since group members usually are interested in subjects that are expressed in the group. Based on our observation, people who are in a same group in a virtual environment will likely have the same interest. However, it is not always true because some people may randomly join a group and most of the time they are not attentive in this group's interest. There should be another factor to measure the level of the degree of belonging of each user to a group, for example, based on the common tags that are used between a user and a group. Hence, if the belonging level of two users to a common group is very high, the probability that the two users are similar to each other increases. To do so, a tag set is considered for each group. Each group contains the users who are interested in this group. Besides, each user in the system has a tag set containing all the tags that she/he assigned to her/his selected items. Each group's tag set contains the assigned tags of all the members. The process of finding each group's tag set is described in Sect. 4. The belonging level of a user u to a group g is defined below.

$$ms(u, g) = \frac{\sum_{ta \in (T_u \cap T_g)} freq(u, ta)}{\sum_{tu \in (T_g)} freq(g, tu)} \tag{3}$$

where *ta* represents a tag in the intersection of tag set of user *u*, *Tu*, and tag set of group *g*, *Tg*. Also, *freq* (*u,ta*) determines the frequency of *ta* by user *u*. Moreover, *tu* represents a tag of *tu* which belongs to tag set of group *g*.

Finally, in order to calculate the similarity of user *u* and *v* based on their membership information, Eq. (4) is defined.

$$MSim_{u,v} = \frac{\sum_{g_i \in (G_u \cap G_v)} ms(u, g_i) * ms(v, g_i)}{|G_u \cap G_v|} \tag{4}$$

where *Gu* is a set of groups that user *u* joined, *gi* represents a group which is in the intersection of group sets of user *u* and group sets of user *v*, which means *gi* is a shared group between *u* and *v*, and $Gu \cap Gv$ is a set of all shared groups between users *u* and *v*.

3.1.4 Overall Similarity

As mentioned before, the ultimate goal of this paper is to fuse the social networking information such as friendship and membership into the collaborative filtering algorithm, in order to enhance the accuracy of recommendations. To do so, firstly, $TSim_{u,v}$ is calculated and then the amplifying method is applied in order to take into account the friendship for those who have a strong relationship with a target user. Afterwards, membership information is incorporated for the purpose of taking advantage of the shared interests in a group.

$$Sim_{u,v} = \alpha * TSim_{u,v} + (1 - \alpha)\left(\beta * FSim_{u,v} + (1 - \beta) * MSim_{u,v}\right) \tag{5}$$

In order to compute the overall similarity we have defined two parameters α and β to adjust the weight of different factors.

In Eq. (5), α and β are applied with the intention of regulating the weight influence of implicit and explicit relationships. The precise value of these terms should be determined empirically. The process of finding α and β is described in Sect. 4. To keep the overall similarity value between 0 and 1 we consider $0 < \alpha, \beta < 1$.

In Eq. (5), α is applied to adjust the weight between item-tag-based similarity ($Tsim_{u,v}$) and the social networking information. Then β adjusts the relative weights between these two types of social relationships which are the similarity based on friendship ($FSim_{u,v}$) and the membership-based similarity ($MSim_{u,v}$). In this sense, the bigger α is, the greater the weight of the tagging activity is. In doing so, tagging activity plays a more important role. On the other hand, a bigger β value implies that the friendship-based similarity plays a more important role in the overall

similarity. Since each system has different features, for example, maybe in a system, membership information is more reliable than friendship information, by adjusting these two values, we can determine which factor plays a more important role in our decision about computing the similarity value. After the computation of $Sim_{u,v}$ for finding neighbors, the next step is to recommend items to users by predicting each item's ratings [32]. In brief, in the above equation the similarity of the neighbor user and the given user is computed.

3.2 Making Recommendation

One of the most important steps in recommendation systems is predicting the future behavior of a user. At first, a subset of similar users to a target user based on their similarities is calculated and then the weighted aggregation of their ratings is applied to make recommendations for the user [35].

The next substantial step is predicting the future behavior of a user. Our system provides the target user u with a sorted list of items that she/he will likely to select in the future. The interest level of a user u to a particular item that is selected by neighbor v depends on two main components: the similarity of the neighbor v to the user u and the similarity of item i selected by the neighbor v to the items that are tagged by the user u. In order to find the similarity of items in view of tags that are assigned to them by users, we make use of the weighted Jaccard similarity method [36] with some modifications.

$$SimItem_{i,j} = \frac{\sum_{t \in (v_i \cap v_j)} \mathrm{Min}(v_i(t).Fq, v_j(t).Fq)}{\sum_{t \in (v_i \cap v_j)} \mathrm{Max}(v_i(t).Fq, v_j(t).Fq) + \sum_{ta \in (v_i \cup v_j - v_i \cap v_j)} \mathrm{Max}(v_i(ta).Fq, v_j(ta).Fq)} \tag{6}$$

According to Eq. (6), there is a vector, *vi*, for each item in which each element of this vector is a tag frequency pair representing the tag name and its frequency. Thus, $vi(t).Fq$ represents the frequency of tag t on item i. Finally, based on the user and the item similarity we predict those items that the target user will probably select in the future. The algorithm below demonstrates how the top-N recommendation list for user u is generated.

Input: user u, u's neighbor list, item similarity matrix
Initialization:

i. NIAvgs=**null**; a list that holds objects of:
   Struct NIAvg{ Item *i,* Float avg}
ii. TopNs=**null**; a list that holds objects of:
   Struct TopN{Item *i,* **Float** rank}
iii. ItemInterests=**null;** an array that holds objects of: **Struct** ItemInterest {Item*j*, **Float** IUSim}

Step1: FOREACH *v* as neighbor of *u*
FOREACH item *i* in *v*'s item list
i. **FOREACH** item *j* in *u*'s item list
ItemInterests [*i*].**ADD (***j, SimItem*$_{i,j}$**Sim*$_{u,v}$**);**
ii. NIAvgs.**ADD**(*i,* ItemInterests [*i*].**AVG**()**);**
Step2: FOREACH *gi* as NIAvg.**Groupby**(i)
TopNs.**ADD**(*gi*.**Key,** *gi*.**Max**()**);**
Step3: TopNs.**Sort**();
RETURN TopNs.**Select** (Item);

First, multiplications of item similarity of item list of user *u* and her/his neighbors' item lists and user similarity of *u* and *v* are stored in an array of lists, called *ItemInterests*. Of course, it comes up with various values for each item of neighbor *v*. Thus, each member of this array is considered as a list that holds different duplications of each item of neighbor *v*. In the next step we calculate the average value of duplicated items and store them in *NIAvgs*. In fact, this average value determines how much a particular item *i* is similar to a user's taste. The higher average value, the more chance that target user selects this item in the future. Maybe there is still some duplication of some items in *NIAvgs*. For instance, item *i* is a common item between two neighbors then according to the similarity value of these neighbors and *u*, two different values are calculated for *i*. In doing so, the algorithm finds the maximum value of each item and returns a sorted list of items based on their final values.

The motivation behind combining the item similarity in the item recommendation is explained in the following paragraphs:

(1) *If we recommend items only by considering the user similarity*:
If a nearest neighbor user selects an item, it is not reasonable to recommend this item to the target user only because the nearest neighbor user selected this item. The necessary and sufficient condition for recommending an item to the user could be explained in a way that the neighbor user should select this item and be interested in the selected item.
(2) *If we recommend items only by considering the item similarity*:
If one of the items selected by the nearest neighbor user is very similar to the items of the given user, still there is no guarantee to put this item at top of the recommendation list because although a similar item is selected by the user, maybe he/she is not interested in this item.

Thereupon, we should better consider the similarity of the user with her/his nearest neighbor users combined with the item similarity when recommending items.

4 Evaluation

4.1 Dataset

One of the most important steps in evaluating this class of algorithms is choosing a proper dataset. We selected a dataset with two main features. The first feature is that the dataset covers all of the possible situations (various factors) of the proposed approach. The second feature is that our dataset should be usable for evaluating other similar algorithms in order to evaluate the improvement of our proposed algorithm. Furthermore, we were looking for a dataset that has all the features such as user selected items and their assigned tags, friendship, and membership information. In order to cover all these requirements we used a Last.fm dataset[6] that was gathered in the first half of 2009 which is a popular and standard dataset to evaluate music recommender systems. Last.fm is a music recommender web site that makes a profile for each user based on her/his previously listened-to songs and incorporates social networking information. In the Last.fm people can make friends as well as join their interested groups. Last.fm can predict the most suitable item and recommend it to a user utilizing the collaborative filtering and social network information.

There are many algorithms that are tested using this dataset. Among all of Last.fm's datasets, we selected the one that provides us with the required information. Table 2 presents a comprehensive description of the attributes of our dataset. In this dataset, items are sound tracks that are listened by users. An annotation is the tuple <user_id, item_id, tag_id> demonstrating a user, her/his listened track and the assigned tag to that track. Groups consists of pairs of <id_group, id_user> presenting a group and a user that is member of this group. Friends are lists of pairs of <id_userA, id_userB> which means user *A* is a friend of user *B* and vice versa.

Table 2 Dataset characteristics

Dataset feature	Numbers
Users	99,405
Annotations	10,936,545
Items	1,393,559
Tags	281,818
Friends	66,429
Groups	1,048,576

[6] http://www.di.unito.it/~schifane/dataset_lastfm_WSDM.zip.

4.2 Preprocessing Steps

In our evaluation, two adopted standard evaluation metrics precision [37] and recall [38] are used to measure the accuracy of our algorithm. The following equations show how to calculate the precision and recall.

$$p_u = \frac{|hits_u|}{|recSet_u|} \tag{7}$$

$$R_u = \frac{|hits_u|}{|testSet_u|} \tag{8}$$

where Pu donates the precision value for user u, $|hitsu|$ represents the number of correctly recommended items to user u, $|recSetu|$ is the total number of recommended items to user u, Ru is the recall value for user u and $|testSetu|$ the number of items in the test set of user u. In doing so, the high precision value illustrates that more relevant than irrelevant outcomes are returned. Also, the high recall value explains that most of the relevant outcomes are returned. Therefore, based on precision and recall metrics we evaluate the accuracy of our algorithm.

4.2.1 Finding the Tag Set of Each Group

For this step 10 % of our dataset is selected. In our implementation, in order to find the tag frequency in each group's tag set, we only consider tags which occur very frequently. The reason for removing some tags is that there might be some tags that are assigned to a group only a few times, and these tags cannot represent the common interests of group members. We consider them as outliers. For each user there is a tag frequency list which contains objects in pairs of Tag-ID and its frequency. Similarly, there is a tag frequency list for each group which is generated from all of the tag frequency lists of users who are members of this group. We sort the tag frequency list of each group. Then, the top 50 % tags of this sorted tag frequency list are considered in our further calculation.

4.2.2 Finding α and β

In order to achieve the best performance of our approach some preprocessing in terms of initialization of some variables is needed. According to Eq. (5), a suitable scale for both α and β is between 0 and 1. Figure1 illustrates the possible values of the combination of α and β in a square with the length of 1. In other words, several combinations of α and β will fit in this square.

In the direction of discovering the most appropriate value of α and β, we vary these values in an increment of 0.1 to find the best combinations of α and β, which is the combination that has the highest precision value. The highest precision value

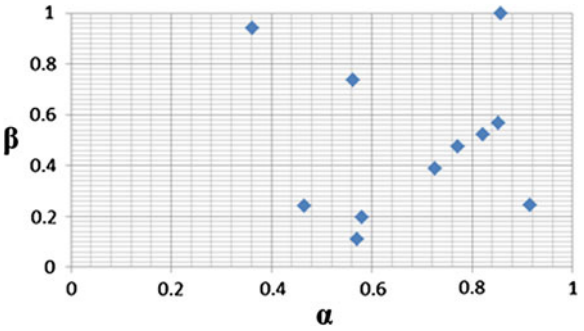

Fig. 1 Possible combinations of α and β values

Table 3 Top 20 precision values for different α and β values

	α								
β	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	18.03	18.28	19.47	20.72	20.59	19.32	18	19.62	19.16
0.2	19.73	20.51	19.02	19.24	20.1	20.09	19.92	20.03	20.89
0.3	20.22	19.42	20.82	21.12	19.01	21.1	20.92	19.22	20.45
0.4	19.35	20.69	21.24	20.92	21.06	21.13	21.34	20.65	19.19
0.5	21.64	18.82	19.99	21.27	21.08	22.06	19.51	19.96	20.31
0.6	21.34	19.51	18.97	21.47	21.31	22.15	20.28	21.97	18.49
0.7	19.22	19.76	20.11	21.09	22.21	22.56	22.19	19.3	19.83
0.8	19.89	20.44	18.53	18.7	21.18	21.54	22.38	20.01	19.87
0.9	18.4	19.28	19.26	19.08	21.84	21.04	20.54	21.75	18.35

indicates that most of the relevant items are returned to the user which shows the effectiveness of the proposed approach. In Table 3, we examined the precision values based on Eq. (5) while returning the Top 20 recommendations.

Table 3 shows the performance of our algorithm reaches its peak when α is 0.6 and β is 0.7. This means that the tagging activity contributes 60 %, the friendship and membership relations contribute 28 and 12 %, respectively, in the overall similarity calculation. We have a similar result for Recall. So these will be the final values we use for the later experiment.

4.3 Experiment Results

In order to assess both error rate and cumulative performance of the proposed algorithm, 80-20 method is applied [39]. In the 80-20 testing method, 80 % of the dataset is selected randomly as the training set and the remaining 20 % of the dataset is selected as the testing set. The recommender system recommends a list of ordered items to the test user which has not been selected by this user before. According

to the training set information we predict the interest probability of users to those items which are not selected by users. In the test set we have the information of users and their selected items. Thus, in the evaluation part we check for each user if the recommended items based on the training set are the same as the items selected by the user in the test set or not.

In details, based on the prediction scores which are sorted in a descending order, the ordered Nlist which we propose in the recommendation algorithm will be recommended to the user. If the test user has already selected or tagged the recommended item which is in the Nlist, then the item would be counted as a hit. Therefore, for each group of test user's dataset we compute the average precision and recall. These numbers are used to measure the accuracy of the recommendation algorithm. Tables 4 and 5 show the precision and recall results of our proposed approach.

The first row (Sim_{TUI}) shows the results when the similarity is only based on item-tag activity. The second row ($Sim_{TUI+fri}$) shows the results when the nearest neighbors are computed considering tagging activity information of users and their friendship relations. In fact, combining friendship information with the tagging activity information is useful while the algorithm returns 2 items. The results in the third row ($Sim_{TUI+mem}$) illustrate that membership information are beneficial in returning 5 items, while the nearest neighbors are calculated based on the tagging activity of users and their membership information. The results in the last row ($Sim_{TUI+fri+mem}$) demonstrate that when the number of recommended items are increased, it is more possible that the recommended items are the user desired items.

With the purpose of evaluating the effectiveness of our proposed algorithm, our study is compared with one of the most closely related algorithm to our approach which combines membership and friendship information with CF via weighted similarity approach [32] which is called Augmenting algorithm in this paper.

From the results shown in Table 6 for the augmenting algorithm, when we add the social information such as friendship or membership, the precision value could

Table 4 Precision when using our proposed algorithm

Precision	Top 1	Top 2	Top 5	Top 10	Top 20
Sim_{TUI}	30.15	35.22	26.27	24.27	18.85
$Sim_{TUI+fri}$	28.48	38.02	27.68	23.78	19.51
$Sim_{TUI+mem}$	27.17	33.38	31.1	24.41	19.57
$Sim_{TUI+fri+mem}$	29.49	32.76	29.32	30.95	21.92

Table 5 Recall when using our proposed algorithm

Recall	Top 1	Top 2	Top 5	Top 10	Top 20
Sim_{TUI}	6.68	11.03	21.63	34.19	31.61
$Sim_{TUI+fri}$	6.63	12.66	24.52	37.27	31.12
$Sim_{TUI+mem}$	5.94	12.45	25.81	35.25	31.86
$Sim_{TUI+fri+mem}$	5.78	12.40	24.91	38.43	35.17

Table 6 Precision when using the augmenting algorithm

Precision	Top 1	Top 2	Top 5	Top 10	Top 20
Sim_{UI}	29.47	27.21	23.2	17.23	10.45
Sim_{UI+fri}	29.86	30.3	28.94	18.21	12.45
Sim_{UI+mem}	29.85	36.83	28.84	18.29	12.35
$Sim_{UI+fri+mem}$	29.63	33.78	28.75	20.75	11.3

always be improved compared to the case when we only consider the tag information. Friendship information normally could provide a better result than the membership information. Combining three of them achieves the best result for recommending Top 10 items.

The results of recalls for the augmenting algorithm are shown in Table 7. We can get similar conclusion on the recall value. Combining the tagging activity of the user with the social information can also improve the performance on recall values.

If we compare the last rows of Tables 4 and 6, or Tables 5 and 7, it can be seen that our algorithm is more accurate than the Augmenting algorithm while returning 5, 10, and 20 items.

We further show the comparison in Figs. 2 and 3. According to the two figures, in Top 1, Top 2 and Top 5 in some situations of combining via friendship and

Table 7 Recall when using the augmenting algorithm

Recall	Top 1	Top 2	Top 5	Top 10	Top 20
Sim_{UI}	5.95	11.45	20.9	28.95	26.15
Sim_{UI+fri}	6.84	13.10	24.69	30.61	25.83
Sim_{UI+mem}	5.32	11.37	24.56	29.47	23.16
$Sim_{UI+fri+mem}$	6.49	11.97	24.58	30.7	24.39

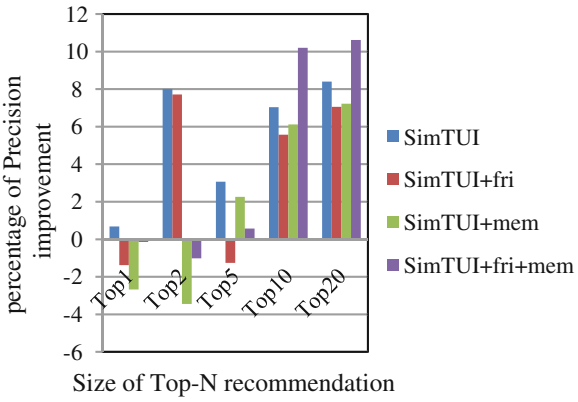

Fig. 2 Improvements on precision for all of the fusion approaches

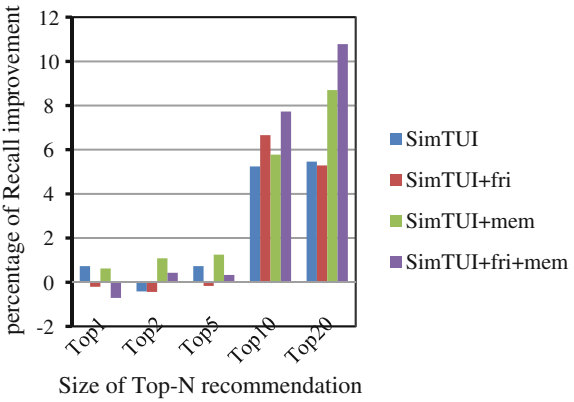

Fig. 3 Improvements on recall for all of the fusion approaches

membership the improvements are not tangible. However, combining the tagging activity with the friendship and membership information in Top 10 and Top 20 causes an improvement in our results.

5 Conclusions

Social tagging systems provide recommendations to users based on what tags other users have used on items. We developed a similarity metric, based on social tagging information, to model three types of relationships: users tagging, friendship, and membership. Moreover, we have proposed a new recommendation method, which applies user similarity to find the most interesting items to target user's taste, and it also takes item similarity into consideration to sort the recommended items.

In this paper we tried to separate the influential friends of the target user (those friends whose tastes are more similar to the target user's taste) from the noninfluential friends of the target user (those users whose tastes are not similar to the target users' taste). Also, the membership information was useful in finding the similarity of users based on the two factors: their shared groups and their belonging level to those shared groups. Consequently, the nearest neighbors of the target user were found by combining implicit relations (similarity of users based on common tags on common items) with explicit relations (similarity of users based on their friendship and membership relations). Furthermore, to recommend items, we considered item similarity as well as user similarity scores. To the best of our knowledge, the work is one of the first efforts which combine the similarity of users based on their shared tags on shared items with their similarity based on friendship and membership information, and also recommends the items by considering the user similarity and the item similarity. Our experimental results show that our proposed approach is effective.

As a further line of research, it would be extremely interesting to study the use of the semantic information of those tags. Thus, we may extend our approach to a novel semantic-based method with a hybrid approach which applies combination of CF and the content-based filtering to check if it could further improve the performance. It means that we need to analyze the semantic meaning and context of social tags to find the similar users or similar items [40]. Another interesting direction is to apply the inverse user frequency (IUF) concept which assumes that generally liked items are less important in similarity computing than the less common items.

Although recommender systems provide impressive solutions for recommending preferred resources to users, these techniques fail to evaluate the fluctuating behavior of users [1, 34]. Another direction we would like to consider is to extend our approach to integrate the time dimension as a measure to assess the importance of an item-tag pair. Moreover, we can push forward the use of the friendship relations by considering the transitive relationship (friend of friend) between users. Therefore, a new graph can be created which defines more broad relationships between users.

References

1. Rafeh R, Bahrehmand A (2012) An adaptive approach to dealing with unstable behaviour of users in collaborative filtering systems. J Inf Sci 38:205–221
2. Agresti A, Winner L (1997) Evaluating agreement and disagreement among movie reviewers. Chance 10:10–14
3. Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the ACM conference on computer supported cooperative work, pp 175–186
4. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst (TOIS) 22:5–53
5. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17:734–749
6. Sarwar B, Karypis G, Konstan J, Riedl J (2000) Application of dimensionality reduction in recommender system-a case study. In: Proceedings of the ACM WebKDD 2000 web mining for E Commerce workshop
7. Goldberg K, Roeder T, Gupta D, Perkins C (2001) Eigentaste: a constant time collaborative filtering algorithm. Inf Retr 4:133–151
8. Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques. Adv Artif Intell 2009:1–20
9. Miller BN, Konstan JA, Riedl J (2004) PocketLens: toward a personal recommender system. ACM Trans Inf Syst (TOIS) 22:437–476
10. Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the fourteenth conference on uncertainty in artificial intelligence, pp 43–52
11. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on world wide web, pp 285–295
12. Durao F, Dolog P (2009) A personalized tag-based recommendation in social web systems. Adapt Personal Web 2:40

13. Cho YH, Kim JK, Kim SH (2002) A personalized recommender system based on web usage mining and decision tree induction. Expert Syst with Appl 23:329–342
14. Tso-Sutter KHL, Marinho LB, Schmidt-Thieme L (2008) Tag-aware recommender systems by fusion of collaborative filtering algorithms. In: Proceedings of the 23rd ACM symposium on applied computing, pp 16–20
15. Wu C, Zhou B (2009) Analysis of tag within online social networks. In: Proceedings of the ACM 2009 international conference on supporting group work, pp 21–30
16. Givon S, Lavrenko V (2009) Predicting social-tags for cold start book recommendations. In: Proceedings of the third ACM conference on recommender systems, pp 333–336
17. Parra D, Brusilovsky P (2009) Collaborative filtering for social tagging systems: an experiment with CiteULike. In: Proceedings of the third ACM conference on recommender systems, pp 237–240
18. Sen S, Vig J, Riedl J (2009) Tagommenders: connecting users to items through tags. In: Proceedings of the 18th international conference on world wide web, pp 671–680
19. Kazienko P, Musial K, Kajdanowicz T (2011) Multidimensional social network in the social recommender system. IEEE Trans Syst Man Cybern Part A: Syst Hum 41:746–759
20. Perez LG, Montes-Berges B, del Rosario Castillo-Mayen M (2011) Boosting social networks in Social Network-Based Recommender System. In 11th international conference on intelligent systems design and applications (ISDA), pp 426–431
21. Yu-Shian C, Kuei-Hong L, Jia-Sin C (2011) A social network-based serendipity recommender system. In: International Symposium on intelligent signal processing and communications systems (ISPACS), pp 1–5
22. Ziegler C-N, Lausen G (2004) Analyzing correlation between trust and user similarity in online communities. In: Trust management, Springer, Heidelberg pp 251–265
23. Avesani P, Massa P, Tiella R (2005) A trust-enhanced recommender system application: moleskiing. In: Proceedings of the ACM symposium on applied computing, pp 1589–1593
24. Golbeck J (2006) Generating predictive movie recommendations from trust in social networks. In: Proceedings of the 4th international conference on trust management, iTrust, pp 93–104
25. Massa P, Avesani P (2007) Trust-aware recommender systems. In: Proceedings of the ACM conference on recommender systems, pp 17–24
26. Liu F, Lee HJ (2010) Use of social network information to enhance collaborative filtering performance. Expert syst appl 37:4772–4778
27. Ye M, Liu X, Lee W-C (2011) Exploring social influence for recommendation-a probabilistic generative model approach, arXiv preprint 1109.0758
28. Konstas I, Stathopoulos V, Jose JM (2009) On social networks and collaborative recommendation. In: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval, pp 195–202
29. Vasuki V, Natarajan N, Lu Z, Dhillon IS, Affiliation recommendation using auxiliary networks. In: Proceedings of the fourth ACM conference on recommender systems, pp 103–110
30. Chen W-Y, Chu J-C, Luan J, Bai H, Wang J, Chang EY (2009) Collaborative filtering for orkut communities: discovery of user latent behavior. In: Proceedings of the 18th international conference on world wide web, pp 681–690
31. Spertus E, Sahami M, Buyukkokten O (2005) Evaluating similarity measures: a large-scale study in the orkut social network. In: Proceedings of the eleventh ACM SIGKDD international conference on knowledge discovery in data mining, pp 678–684
32. Yuan Q, Zhao S, Chen L, Liu Y, Ding S, Zhang X et al (2009) Augmenting collaborative recommender by fusing explicit social relationships. In: Workshop on recommender systems and the social web, recsys, pp 46–56
33. Osatapirat K, Limpiyakorn Y (2011) Capturing personal direct and indirect interests in social tagging systems using virtual tag space. In: International conference on future information technology IPCSIT, pp 249–253
34. Zheng N, Li Q (2011) A recommender system based on tag and time information for social tagging systems. Expert Syst Appl 38:4575–4587

35. Herlocker JL, Konstan JA, Borchers A, Riedl J (1999) An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval, pp 230–237
36. Braunhofer M, Kaminskas M, Ricci F (2011) Recommending music for places of interest in a mobile travel guide. In: Proceedings of the fifth ACM conference on recommender systems, pp 253–256
37. Huang Z, Zeng D, Chen H (2007) A comparison of collaborative-filtering recommendation algorithms for e-commerce. Intell Syst IEEE 22:68–78
38. Dietmar Jannach MZ, Felferning A, Friedrich G (2011) Recommender systems: an introduction. Cambridge University Press, New York
39. Sarwar B, Karypis G, Konstan J, Riedl J (2000) Analysis of recommendation algorithms for e-commerce. In: Proceedings of the 2nd ACM conference on electronic commerce, pp 158–167
40. Fernández-Tobías I, Cantador I, Bellogín A (2012) Semantic disambiguation and contextualisation of social tags. LNCS 7138:181–197

Personalization of Web Search Using Social Signals

Ali Khodaei, Sina Sohangir and Cyrus Shahabi

Abstract Over the last few years, Web has changed significantly. Emergence of social networks and Web 2.0 have enabled people to interact with Web document in new ways not possible before. In this paper, we present *PERSOSE* a new search engine that personalizes the search results based on users' social actions. Although the users' social actions may sometimes seem irrelevant to the search, we show that they are actually useful for personalization. We propose a new relevance model called persocial relevance model utilizing three levels of social signals to improve the Web search. We show how each level of persocial model (users' social actions, friends' social actions and social expansion) can be built on top of the previous level and how each level improves the search results. Furthermore, we develop several approaches to integrate persocial relevance model into the textual Web search process. We show how *PERSOSE* can run effectively on 14 million Wikipedia articles and social data from real Facebook users and generate accurate search results. Using PERSOSE, we performed a set of experiments and showed the superiority of our proposed approaches. We also showed how each level of our model improves the accuracy of search results.

Keywords Social search · Personalized search · Social network · Facebook · Information retrieval · Wikipedia

A. Khodaei
Yahoo! Corporation, 701 1st Ave., Sunnyvale, CA 94089, USA
e-mail: alik@yahoo-inc.com

S. Sohangir (✉)
GraphDive Company, 1295 El Camino Real, Suite B, Menlo Park, CA 94025, USA
e-mail: sohangir@yahoo.com

C. Shahabi
Department of Computer Science, University of Southern California,
Los Angeles, CA 90089, USA
e-mail: shahabi@usc.edu

Ö. Ulusoy et al. (eds.), *Recommendation and Search in Social Networks*,
Lecture Notes in Social Networks, DOI 10.1007/978-3-319-14379-8_8

1 Introduction

While in early stages, search engines' focus was mainly on searching and retrieving relevant document based on their content (e.g., textual keywords), new search engines, and new studies start to focus on context alongside content as well. For instance, [1] proposed a search engine that combines traditional content-based search with context information gathered from users' activities. More recently, search engines started to make the search results more personalized. With personalized searches, search engines consider the searchers' preferences, interests, behavior, and history. The final goal of personalized search as well as other techniques studying users' preferences and interests is to make the returned results more relevant to what the user is actually looking for.

Emergence of social networks on the Web (e.g., Facebook and Google Plus) have caused the following key changes on the Web. First, social networks reconstruct friendship networks in the virtual world of the Web. Many of these virtual relationships are good representatives of their actual (friendship) networks in the real world. Second, social networks provide a medium for users to express themselves and freely write about their opinions and experiences. The social data generated for each user is a valuable source of information about that users' preferences and interests. Third, social networks create user identifiers (identities) for people on the Web. Users of a social network such as Facebook will have a unique identity that can be used in many places on the Web. Not only such users can use their Facebook identities on the social network itself but they can also use that identity to connect and interact with many other web sites and applications on the Web. Along the same lines, social networks such as Facebook and Google Plus provide utilities for other web sites to get integrated with them directly, enabling users of the social network to interact directly with those web sites and Web documents using their social network identity. For instance, a Web document can be integrated into Facebook (using either *Facebook Connect* or *instant personalization*[1]) allowing every Facebook user to perform several actions (e.g., *LIKE*, *RECOMMEND*, *SHARE*) on that document. Finally, many search engines are starting to connect to social networks and allows users of such social networks to be the users of the search engine. For instance, the Bing search engine is connected to Facebook and hence users with their Facebook identities can log in into Bing to perform their searches.

The above developments inspired us to study a new framework for search personalization. In this paper, we propose a new approach for performing personalized search using users' social actions (activities) on the Web. We utilize the new social information mentioned above (users' social activities, friendships, user identities, and interaction of users on Web documents) to personalize the search results generated for each user. We call this new approach to personalization of search, *persocialized search* since it uses *social* signals to *personalize* the search. While a traditional

[1] https://developers.facebook.com/docs/guides/web/.

personalized search maintains information about the users and the history of their interactions with the system (search history, query logs), a persocialized search system maintains information about the users, their friendships (relations) with other users and their social interactions with the documents (via social actions).

Recently, McDonnell and Ali [2] conducted a complete survey on the topic of *social search* and various existing approaches to conduct social search. As mentioned in McDonnell and Ali [2] there exist several definitions for social search: One definition is the way individuals make use of peers and other available social resources during search tasks [3]. Similarly, Vuorikari et al. [4] defines social search as using the behavior of other people to help navigate online, driven by the tendency of people to follow other people's footprints when they feel lost. A third definition is by Amitay et al. [5] and is defined as searching for similar-minded users based on similarity of bookmarks. Finally, Evans et al. [6]'s definition of social search includes a range of possible social interactions that may facilitate information seeking and sense-making tasks: utilizing social and expertise networks; employing shared social work spaces; or involving social data mining or collective intelligence processes to improve the search process. For us, social search focuses on utilizing querying users' as well as her friends' **social actions** to improve the conventional textual search. By integrating these social actions/signals into the textual search process, we define a new search mechanism: *persocialized search*. Our main goal in this paper is to prove our hypothesis that these social actions (from the querying user and his friends) are relevant and useful to improve the quality of the search results.

Toward this end, we propose a new relevance model called the *persocial* relevance model to determine the social relevance between a user and a document. Persocial model is developed in three levels, where each level complements the previous level. First, we are using social actions of a user on documents as implicit judgment/rating of those documents by the user. For instance if a Facebook user u, performs any type of social action (e.g., LIKES, SHARES) on document d, she implicitly expresses her positive opinion about d. As a result, d should get a slightly higher score for queries relevant to d and issued by u. In Sect. 4, we show that using social actions from each user and boosting documents' score with such actions (level 1), by itself improves the accuracy of search results. Second, it is both intuitive and proven [7] that people have very similar interests with their friends. Also, people tend to trust the opinions and judgements of their friends more than strangers. As a result, not only the documents with direct social actions by user u are relevant to u, but also those documents with social actions performed by u's friends are also relevant to user u. Hence, we adjust (increase) the weights given to those documents for relevant queries issued by u. As we discuss in more details in Sect. 3, many parameters such as the strength of social connections between users as well as the influence of each user must be incorporated in to the model for generating the most accurate results. In Sect. 4, we show that using the social signals from the friends will improve the search results significantly. Furthermore, we show that using a combination of user data and his/her friends data generates the best results. Finally, the Web documents are often well-connected to each other. We argue that social features of each document should be dynamic, meaning that social actions/signals of the document can and should be

propagated to other adjacent documents. A user's interest for a document—shown by a social action such as *LIKE*—can often imply the users' interests in other relevant documents—often connected to the original document. Thus, we use connections among documents to let social scores flow among documents, hence generating a larger document set with more accurate persocial relevance scores for each user.

In sum, the major contribution of this paper is to propose a model and build a system for utilizing users' social actions to personalize the Web search. We propose a new relevance model to capture relevance between documents and users based on users' social activities. We model three levels of personalization based on three sets of social signals and show how each level improves the Web search personalization. In addition, we propose three new ranking approaches to combine the textual and social features of documents and users. Furthermore, we develop a persocialized search engine dubbed *persocialized search engine* ('PERSOSE' for short) to perform persocialized search on real data using real users. Using *PERSOSE*, we conduct a comprehensive set of experiments using 14 million documents of Wikipedia as our document set and real Facebook users as our users. As a result of the experiments, we show that social actions of a user, his friends' social actions and social expansion of documents (all three levels of social signals) improve the accuracy of search results.

2 Overview

In this section, we present the problem statement without going into much details (we present some of definitions/formalizations in Sect. 3.1). We also provide the system overview of *PERSOSE*.

The objective of *PERSOSE* search engine can be stated as follows:

Suppose $D = \{d_1, d_2, \ldots, d_n\}$ is the set of documents that exist in our system. Each document is composed of a set of textual keywords. Also, there is a set $U = \{u_1, u_2, \ldots, u_m\}$ of users interacting with the system. Users can search for documents but more importantly users can also perform a set of defined *social actions* (e.g., *LIKE*, *RECOMMEND*, *SHARE*) on the documents. We also assume a social network modeled as a directed graph $G = (V, E)$ whose nodes V represent the users and edges E represent the ties (relationship) among the users. Finally, each query issued to the system has two parts: the textual part of the query which is presented by a set of textual keywords (terms), and social part of the query which is defined mainly as the user issuing the query. The goal of *PERSOSE* is to first identify and model the social dimension of the documents in the system, and next to score and rank the documents based on their relevance to both the textual and the social dimensions of the query. We call this type of search performed by *PERSOSE*, *PerSocialized Search* since search is *personalized* using *social* signals.

System Overview. A general overview of *PERSOSE* is displayed in Fig. 1. As shown in this figure, there exist two types of objects/modelus in *PERSOSE*: Modules that belong to the (existing) textual search models and modules that are new and are

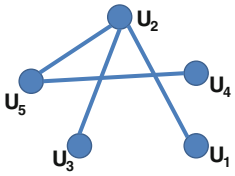

Fig. 1 Overview of *PERSOSE*

part of the new social model. In Fig. 1, textual modules are displayed by solid lines, social modules are depicted by dotted lines and modules with both textual and social features are shown by mixed lines.

Accordingly, *PERSOSE* has two engines: (1) the textual engine reads (crawls) the documents in the system and generates the necessary textual metadata for each document (e.g., textual vectors); there is nothing new about the textual engine, (2) the social engine has two inputs. One is the social network G with all its properties and relationships. The second data structure maintains a dataset of users' social activities. This dataset, for each user in the social network, contains all their social activities (feed) including their interaction with documents in the system. The social engine processes this dataset as well as graph G and generates multiple social vectors for documents and users. In addition to the social vectors, the social engine defines and calculates relevance scores between documents and users as well as among documents. Description of each vector as well as the detailed description of the new relevance model are discussed in Sect. 3.1.

Another major module in our system is the ranker module. Ranker which contains both the textual and persocial aspects, receives queries from each user and generates a ranked list of documents for each query and returns them back to the user. As we mentioned earlier, each query has two parts: the textual part of the query (set of terms) and the user issuing the query. Ranker gets both information as well as different vectors generated from the textual and social engines and using one of the approaches described in Sect. 3.2 ranks the documents based on their (textual and social) relevance to the query. Details of different ranking approaches are discussed in Sect. 3.2.

3 Persocialization

In this section, we show how to personalize the search results using social data or what we call *search persocialization*. First, we propose a new relevance model called persocial relevance model to capture and model the social information for both documents and users. In the second part, we show how to use the proposed persocial relevance model to perform persocialized search and propose various rankings.

3.1 Persocial Relevance Model

In this section, we model social relationships between users and documents as well as other social information about users, and propose a new weighting scheme to quantify the relevance of each user to each document.

We define the persocial relevance model at three levels, each level complementing the previous level. We develop the simplest model in level 1 using minimum amount of social data, i.e., social data from user himself. We extend our model significantly in level 2, creating the core of our persocial model. In this level, we also define multiple new social vectors in order to be able to model the persocial relevance more accurately. In the process of modeling level 2 persocial relevance, we create a new weighting scheme called *uf-ri* weighting scheme and define new weights and weight functions for several relationships in the system. Finally, in level 3, we extend our model even further using the concept of *social expansion.*

3.1.1 Persocial Relevance—Level 1

In the first level of the persocial model, we leverage each user's past social data to calculate the persocial relevance between that user and the documents.

Definition We formalize social interactions between users and documents by *social actions*. We define $A = \{a_1, a_2, \ldots, a_l\}$ as a set of all possible *social actions* available to the system. For each document d_j, a set A_{d_j} defines a set of valid (supported) actions for d_j. A_{d_j} is a subset of A ($A_{d_j} \subseteq A$) and contains all the social actions possible for document d_j. For each user u_i we define a set UDA_i as a set of all document action pairs performed by user u_i. To be more formal, $UDA_i = \{(d_j,a_k) \mid$ if there is an action a_k on document d_j by user $u_i\}$. Each social action is unique and can be applied only once by user u_i on document d_j (nevertheless, that action can be applied by the same user u_i on multiple documents and/or by multiple users on the same document d_j).

Social actions do not have equal importance. We define a weight function W: $A \rightarrow \mathbb{R}$ mapping social actions to real numbers in the range [0, 1]. Values generated by the weight function represent the importance of each social action in the system. The weight function should be designed by a domain expert with the following two constrains: (1) each weight should be between 0 and 1 (inclusive), and (2) the more important the action, the higher the value. The importance of actions are determined based on the domain/application.

Example Assume that our document set contains all the Web pages of a sports web site (e.g., ESPN). Web pages can include news articles, athlete profile pages, sports teams pages and so on. Also, this web site is already integrated (connected) with a social network platform. In this example, all Web pages in our document set are

connected to the Facebook social plug-ins[2] and support the following social actions: *LIKE*, *RECOMMEND* and *SHARE*.

So, $A = \{LIKE, RECOMMEND, SHARE\}$ and also

$A_{d_j} = \{LIKE, RECOMMEND, SHARE\}$ for each and every d_i in our document set (all documents support all actions).

Each user u_i in the system, can *LIKE*, *RECOMMEND* or *SHARE* any document d_j on the web site. With this example, we define weight function W as follows: $W(\text{RECOMMEND}) = 0.6$, $W(\text{LIKE}) = 0.6$, and $W(\text{SHARE}) = 0.8$. These weights indicate that in this domain, *SHARE* is the most important action and *LIKE* and *RECOMMEND* actions have the same importance.

Definition *Persocial relevance—level 1* between document d_j and user u_i is defined based on the number and type of social actions between user u_i and document d_j, and as follows:

$$psRel_{L1}(u_i, d_j) = \sum_{a_k | (d_j, a_k) \in UDA_i} W(a_k)$$

where $psRel_{L1}(u_i, d_j)$ is the persocial relevance level 1 between user u_i and document d_j.

Example In our running example, assume we have two documents d_1 and d_2 and user u_1. User u_1 has *LIKED* and *SHARED* d_1 and he also has *RECOMMENDED* document d_2. Hence, $prRel_{L1}(u_1, d_1) = W(LIKE) + W(SHARE) = 1.4$ and $prRel_{L1}(u_1, d_2) = W(RECOMMEND) = 0.6$.

3.1.2 Persocial Relevance—Level 2

The amount of data generated from one user's social actions is typically insignificant. If we only consider the users' own social actions, many documents will end up having persocial relevance of zero for that user. In addition, as we discussed earlier people have very similar interests with their friends trust the opinions of their friends more than others. Hence, in the second level of persocial model, we utilize friendship relationships between users to improve and extend the level 1 model.

Definition A weight $w_{i,j} > 0$ is associated with each user u_i and document d_j. The term $w_{i,j}$ represents the social importance/relevance of user i to document d and its value is equal to $prRel_{L1}(u_i, d_j)$ defined earlier. For user u_i with no social action on document d_j, $w_{i,j} = 0$.

[2] https://developers.facebook.com/docs/plugins/.

We define *document social vector* to represent the social dimension of the document d_j and represent it as S_{d_j}, defined as bellow:

$$S_{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{m,j})$$

where m is total number of users.

The concept of social vector for a document is analogous (and inspired by) the concept of the textual vector of a document. While textual vector represents the textual dimension of the documents, social vector characterizes the social dimension of the documents. Moreover, our weights $w_{i,j}$ are analogous to term frequency weights ($tf_{i,j}$) in the context of textual search. While each $tf_{i,j}$ indicates the relevance between term (keyword) i and document j, each $w_{i,j}$ represents the relevance between user i and document j. Traditionally (and in the context of textual search), such term frequency is referred as *tf* (*term frequency*) *factor* and offers a measure of how well that term describes the document's textual content. Similarly, we name our social weights ($w_{i,j}$) *uf* (*user frequency*) *factor*. The *uf* factor provides a measure of how well a user describes a document's *social* content.

Example Continuing with our running example, let's add users u_2 and u_3 to the system. Suppose u_2 has *LIKED* document d_1 and u_3 has no social action on d_1. Given this information and previous information about u_1, the social vector for d_1 is as follows.

$$S_{d_1} = (w_{1,1}, w_{2,1}, w_{3,1}) = (1.4, 0.6, 0).$$

Definition We measure $w'_{i,p}$ or the weight between user u_i and user u_p based on the *user relatedness function* between user u_i and u_p. User relatedness function is denoted by $W'(u_i, u_p)$ and measures the relatedness/closeness of two users. There are several existing measures to calculate the relatedness/closeness of two nodes in a graph/social network. Some of the approaches consider the distance between nodes, some look at the behaviors of users in a social network and some take into consideration number of mutual neighbors of two nodes. While the required data is available, any of the above methods or any other existing method can be used for the user relatedness function as long as the following three constraints are satisfied: (1) $W'(u_i, u_i) = 1$, (2) $0 \leq W'(u_i, u_p) \leq 1$ and the more relevant the users, the higher the value, and (3) $W'(u_i, u_p) = 0$ when $W'(u_i, u_p) < \delta$. The first constraint states that each user is the most related user to himself. The second constraint normalizes this measure and also ensures that the more related users are assigned higher scores. Finally, the third constraint filters out all relationships that their significance is below a certain threshold (δ).

Now, we define *user social vector* to represent the social dimension of the user u_i and present it as S'_{u_i}, defined it as below:

$$S'_{u_i} = (w'_{1,i}, w'_{2,i}, \ldots, w'_{m,i}).$$

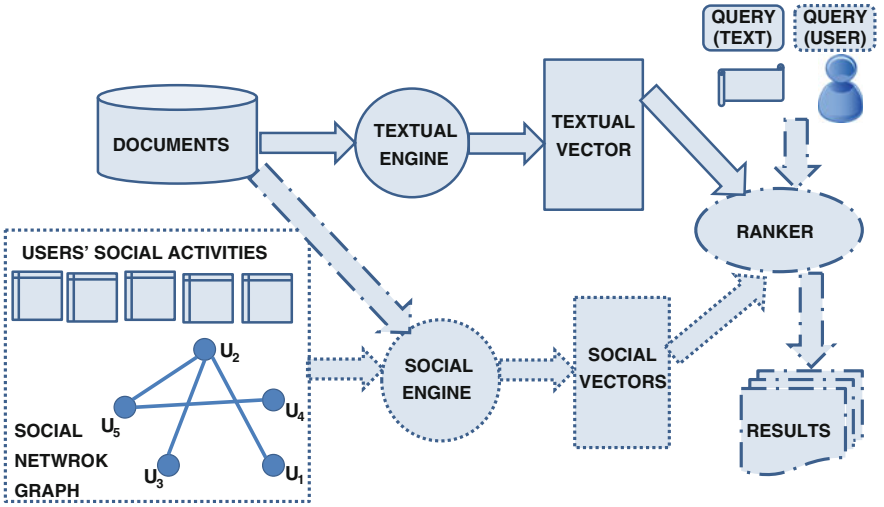

Fig. 2 Friendship structure for the running example

Example Let's add users u_4 and u_5 to the running example. Friendship structure among all five users of our system is depicted in Fig. 2.

In the following, we calculate the user social vector for user u_1 using two different user relatedness functions.

As case 1, we use an inverse of distance between two users (in the network) to capture their relatedness. We also set the threshold value δ equal to 0.3. More formally, $W'(u_i, u_p) = \frac{1}{dist(u_i,u_p)+1}$ where $\delta = 0.3$ and $dist(u_i, u_p)$ is the number of edges in a shortest path connecting u_i and u_p ($dist(u_i, u_i) = 0$). Using this function for user $u1$:

$$\begin{aligned} S'_{u_1} &= (W'(u_1, u_1), W'(u_2, u_1), W'(u_3, u_1), W'(u_4, u_1), W'(u_5, u_1)) \\ &= (1, 0.5, 0.33, 0, 0.33). \end{aligned}$$

Note that $W'(u_4, u_1) = 1/(1 + 3) = 0.25$ but since $0.25 < 0.3$, this value becomes zero.

In addition to relatedness between users, knowing the overall importance/influence of each user also can help us in detecting (and thus giving more weight) to social actions with higher quality and more reliability. Often, when a high profile user (super user) performs a social action on a document, that action and consequently that document are of higher value/quality compared to the case when the same action is performed on the same document by a less influential user.

We quantify the overall (global) importance of each user by the *user weight function* $W''(u_i)$. This measure quantifies the significance of a user in the social network. For instance, with Twitter, a user with many followers will be assigned a higher weight than a user with only few followers, or with Facebook, a user with

more friends is often more important to the social network than a user with fewer friends. In the field of graph theory and social networks, this value is called *centrality* and there exist several approaches to measure it. Four popular methods to compute the centrality value are: degree centrality, betweenness, closeness, and eigenvector centrality [8]. Similar to the user relatedness function, the user weight function is also generic enough and most of the existing approaches can be applied to obtain W''.

Definition We define a weight function $W'': U \rightarrow \mathbb{R}$ mapping users to real numbers in the range [0,1]. Each value $w''(i)$ generated by this weight function represents the overall importance of each user i in the system. The weight function should satisfy the following two constrains: (1) each $w''(i)$ should be between 0 and 1 (inclusive), and (2) the more important the user, the higher the value. The importance of users are determined by user weight function[3] W".

In the context of textual search, there is the *idf* (*inverse document frequency*) *factor* for each term in the system offering a measure to how important (distinctive) is that term in the system. Analogously, we name the weights generated by the weight function W", *ui* (*user influence*) *factor*. The value of *ui* for a user provides a measure of how important that user is in the system.

We define *influence social vector* to represent the importance/influence of all the users, and present it as S''. S'' is defined as follows:

$$S'' = (w''_1, w''_2, \ldots, w''_m).$$

Example For the network depicted in Fig. 2, we use the degree centrality of nodes (users) as an indication of their importance as follows:

$$W''(u_i) = \frac{deg(u_i)}{m-1}$$

where $deg(u_i)$ is the number of edges of node u_i and m is number of nodes (users).

Using the above user weight function, the following weights are generated for the five users:

$$w''(u_1) = 0.5, w''(u_2) = 0.75, w''(u_3) = 0.5, w''(u_4) = 0.25, w''(u_5) = 0.5.$$

Thus, $S'' = (0.5, 0.75, 0.5, 0.25, 0.5)$

Definition *Persocial relevance—level 2* between document d_j and user u_i is defined based on the number and type of social actions between user u_i and document d_j, the relationships between user u_i and other users, the overall importance of each user and the number and type of social actions between user u_i's friends[4] and document d_j, as follows:

[3] Commercialized and more complicated examples of this measure include Klout (klout.com) and PeerIndex (peerindex.com).

[4] To be more precise, set U′ of users such that $\forall u'_l \in U' | W'(u'_l, u_i) > \delta$.

$$psRel_{L2}(u_i, d_j) = \sum_{k=1}^{m} (w(k, j) \times w'(k, i) * \times w''(k)) \quad (1)$$

where $w(k, j)$ is the user frequency (*uf*) factor, $w'(k, i)$ is the user relatedness (*ur*) factor, and $w''(k)$ is the user influence (*ui*) factor. We call this weighting scheme *uf-ri* (user frequency-relatedness influence) weighting scheme. While in classical textual weighting schemes such as tf-idf, for given terms, more weight is given to the documents with (1) more occurrences of terms (tf) , and (2) more important terms (idf), in our *uf-ri* weighting scheme, for a given user, more weight, is given to the documents with (1) more important actions (2) performed by more important users (3) whom are more related (closer) to the given user.

Example Given the values we have so far (using case 2 for W'), the persocial relevance level 2 between $u1$ and document d_1 is calculated as follows:

$$prRel_{L2}(u_1, d_1) = \sum_{k=1}^{5} (w(k, 1) \times w'(1, k) * \times w''(k)) = 1.4 \times 0.5 + 0.6 \times 0.5 \times 0.75 + 0 + 0 + 0 = 0.7 + 0.225 = 0.925$$

3.1.3 Persocial Relevance—Level 3

In this section, we present the concept of *social expansion* and discuss how it can be useful in generating more accurate persocial relevance scores. We show how to define level 3 of persocial relevance by integrating social expansion to the persocial relevance level 2.

Each document on the Web is often well connected to other documents, most commonly using hyperlinks. We argue that social features of each document should be dynamic, meaning that social actions/signals of the document can and should be propagated to other adjacent documents. A user's interest for a document—shown by a social action such as *LIKE*—can often imply the users' interests in other relevant documents—often connected to the original document. In simpler words, we enable social signals to *flow* in the network of documents.[5] We propose to propagate social actions from one document—with some social action—to all documents connected to that document. As an example, imagine a user *LIKES* ESPN's *Los Angeles Lakers* page. Using this signal (action) alone can help us deriving the fact that this document is socially relevant to this user. However, we can do much better by taking into consideration the adjacent documents to the *Los Angeles Lakers* document.

By looking at documents that the original document links to, we can retrieve a new set of documents that are also socially relevant to the user. For our example, the *Los Angeles Lakers* document has outgoing links to document on *NBA* and *Kobe*

[5] Many existing approaches and definitions can be used to measure *connections* between documents. Here, we do not go into details of such approaches.

Bryant. Assuming there is one outgoing link for each of the two documents, half of the original social score can be given to each of these two new documents. As a result, documents on *NBA* and *Kobe Bryant* become socially relevant to the user as well (note that the original *Los Angeles Lakers* document is still more socially relevant to the user than the other two documents.). If we continue this propagation, many new documents will get adjusted social scores from the same social action.

We define *persocial relevance level 3* ($psRel_{L3}$) between document d_j and user u_i as follows:

$$psRel_{L3}(u_i, d_j) = psRel_{L2}(u_i, d_j) + \sum_{d_k \in D_{d_j}} V'(d_k, d_j) \times psRel_{L2}(u_i, d_k) \quad (2)$$

where $psRel_{L2}(u_i, d_j)$ is the persocial relevance between document d_j and user u_i (level 2) as defined in Eq. 1, D_{d_j} is a set of documents connected to the document d_j, and $V'(d_k, d_j)$ is value of *document relatedness function* between document d_k to document d_j. *Document relatedness function* is measuring the connectivity of two documents. Again, we intentionally define this function as generic as possible and do not limit our model by any particular implementation. Simple models like number of hyperlinks between two documents or more sophisticated models such as those that calculate the textual and/or topical similarities between two documents can be used.

The main advantage of using social expansion is to find more *socially relevant* documents for each user. Social expansion also helps in adjusting documents' scores and assigning more accurate relevance scores to each document. Imagine a user who has two explicit *LIKES* on *Google* and *Microsoft*. The same user also has other social actions on *XBOX* and *Bing*. Without using expansion, both *Google* and *Microsoft* generate the same social weight for this user, while using expansion will propagate some weight from both *XBOX* and *Bing* to *Microsoft* and hence gives *Microsoft* a slight advantage (assuming there are links from *XBOX* and *Bing* to *Microsoft*). Using social expansion is also very practical for the current state of the Web where social actions are not very common yet and many documents do not have any social action. Social expansion will help more documents to get scored and hence it will improve the overall social search experience.

3.2 Persocialized Ranking

As described earlier, goal of the ranker module in *PERSOSE* is to personalize and rank the search results using both the social and textual features of the documents. In this section, we discuss three different approaches to rank the documents based on the combination of the textual relevance and persocial relevance scores. In any of the discussed approaches, persocial relevance model of any level (1 through 3) can be applied. Hence, for instance, if friends' information do not exist in the system and

only querying users' own actions are available, we can use persocial relevance level 1 as the persocial relevance model in the proposed approaches. We also incorporate textual relevance in the proposed approaches. Any existing textual model (e.g., tf-idf [9], BM25 [10]) can be used to calculate the textual relevance scores. Furthermore, we have to note that most of the existing search optimization techniques (e.g., pageRank [11]) or other personalized approaches are orthogonal to our approaches and can be added to textual relevance model part (for instance combination of the tf-idf and pageRank can be used as the textual model).

3.2.1 Textual Filtering, Persocial Ranking

In *textual filtering, persocial ranking* (TP) approach, first a regular textual filtering is conducted and all the documents with textual relevance larger than 0 are returned (in the simplest case, documents that have at least one of the query keywords). Next, the remaining documents are scored and ranked using their persocial relevance to the querying user. This is a two-step process in which filtering is based on the textual dimension of the documents and ranking is based on the social aspect of the documents.

3.2.2 Textual Ranking, Persocial Filtering

In *persocial filtering, textual ranking* approach, any document d_j with no persocial relevance to the querying user u_i (i.e., $psRel(u_i, d_j) = 0$) is pruned first. The result of this step is a set of documents with at least one social action from the querying user or her friends (related users). Next, the regular textual search is performed on the remaining documents and documents are scored and ranked based on the textual relevance model. This is also a two-step process with filtering step based on social dimension of the documents and ranking step based on the textual features of the documents.

3.2.3 Persocial–Textual Ranking

With *persocial–textual ranking* approach, both textual and social features of the documents are used simultaneously to calculate the final relevance of the query to each document. We define $Rel(q, d_j)$ as the overall (textual plus persocial) relevance of document d_j with query q. The value of $Rel(q, d_j)$ is defined by a monotonic scoring function F of the textual relevance and persocial relevance values. In *PERSOSE*, F is the weighted sum of the persocial relevance and textual relevance scores:

$$\begin{aligned} Rel(q, d_j) &= F(psRel(u_q, d_j), texRel(T_q, d_j)) \\ &= \alpha.psRel(u_q, d_j) + (1 - \alpha) \times texRel(T_q, d_j) \end{aligned}$$

where T_q is the textual part of the query, u_q is the querying user (social part of the query), $texRel(T_q, d_j)$ is a textual relevance model to calculate the textual relevance between T_q and document d_j, and α is a parameter set by the querying user, assigning relative weights to persocial and textual relevance values.

In this approach and using the above formula, ranking is calculated using both the textual and social features of documents and the query. This is a one-step process with no filtering step.

4 Experimental Evaluation

In this section, we evaluate the effectiveness of *PERSOSE* using data from Facebook and Wikipedia. We first discuss the dataset, approaches and other settings used for the experiments, and then present the results.

Data. For a complete and accurate set of experiments, we need a dataset that contains the following data: (1) a large set of documents with textual information, (2) link structure between documents, (3) real users with friendship relationships, and (4) social actions from users on documents.

Unfortunately no such dataset exists. As a result, we built such a dataset to be used in *PERSOSE* and to evaluate our approaches.

As outlined in Sect. 2, two main data types are fed into *PERSOSE*. One is a set of documents and the other is the social data containing social actions from users as well as relationships among users. We used Wikipedia articles as our document set and Facebook as our social platform. We developed a Web crawler to crawl around 14 million Wikipedia articles and extract textual information from those documents. While crawling, we also captured the relationships among documents and built a (partial) Wikipedia graph. In this graph, each node represents a Wikipedia article. Node d_1 has a directed edge to node d_2 if their Wikipedia articles are related to each other, either explicitly when article d_1 has a link to article d_2, or implicitly when article d_2 is mentioned several times by article d_1. The weight of each connection is based on the frequency and the position of the mentions of one article inside another. The total weight of all outgoing edges for each node of the graph always adds up to one.

As far as the social data, we integrated *PERSOSE* to Facebook using *Facebook Connect*, hence allowing users to log in into *PERSOSE* using their Facebook account and information. When a user connects to *PERSOSE*, our system asks for the permission to read and access users' facebook data. The Facebook data that our system read include users's Facebook activities (e.g., *STATUS*, *LIKES*, *PHOTOS*) as well as users' friendship information. We also read all public data from the users' friends.

Finally, we map users' Facebook activities to social actions on Wikipedia documents. In order to perform this step, we utilized the technology developed at GraphDive[6] to link Facebook data to Wikipedia articles. With GraphDive API, each

[6] http://graphdive.com/.

Facebook activity/post (e.g., *STATUS*, *CHECK-IN*, *LIKE*) can be mapped to one or more than one Wikipedia article. GraphDive algorithm works as follows. GraphDive API receives a Facebook post, parses the text to all possible word-level n-grams ($1 \leq n \leq$ total number of words in the post) and then looks for a Wikipedia article with the same title for each n-gram. For instance, for a *status update* of "I love Los Angeles and Southern California", GraphDive API, will match Wikipedia articles on *Los Angeles*, *California*, and *Southern California* to the post. There are other optimizations taken place by GraphDive API (e.g. disambiguation, varying weights for each n-gram, etc.) that are not the focus of this paper. We only use GraphDive API to map Facebook actions to Wikipedia articles and hence generating a rich set of documents with social actions from real users.

Actions. From the data that Facebook provides via its graph API,[7] we considered the following six actions: *LIKE*, *check-in*, *STATUS*, *PHOTO*, *WORK* and *SCHOOL*. *LIKE* is when a user *likes* a page/topic on Facebook or a document on the Web. *check-in* is when a user *check-ins* his/her location using Facebook. *STATUS* is a free format text usually describing users' activities and feelings. *PHOTO* is a text associated with each photo a user uploads to Facebook. Finally, *WORK* and *SCHOOL* are information about users' workplace and school/university, respectively. Each of the above six actions contain some textual content. As described above, using GraphDive technology, we map those textual content to a set of Wikipedia articles—when possible. For instance, when a user check-ins at *Peet's Coffee and Tea*, using GraphDive, we extract action *check-in* between the user and the Wikipedia article on *Peet's Coffee and Tea,* between the user and the Wikipedia article on *coffee,* and between the user and the Wikipedia article on *tea*.

Approaches. We use three main approaches described in Sect. 3.2 to generate the results: *textual filtering, persocial ranking* (TP), *persocial filtering, textual ranking* (PT) and *persocial–textual ranking* (HB).[8] We also use a baseline approach called *BS*. The *BS* approach generates the results based on the combination of tf-idf and PageRank models.

The same baseline approach is used as the textual model in our existing approaches (whenever textual model needed). The default setting is as follows. The social actions have the same weight (all equal to 0.5) and number of results returned for each approach is 5. When using friends data, we only access data from the top 25 friends (ranked by their user relatedness score to the user) of the user. Also, all four approaches use *expansion* as described in Sect. 3.1.3. Finally, α is set to 0.7 for the *HB* approach (to give more importance to the social part of the search and hence evaluate the impact of social signals more thoroughly). In addition to the main approaches and the baseline approach, we also implemented three levels of the persocial mode on the *hybrid* approach to study and evaluate the impact of each level. Three variations are called: *HB-Level1*, *HB-Level2*, and *HB-Level3*.

Queries. We generate two set of queries for our experiments. The first set called *qset1*, is generated from Google top 100 queries in 2009. For each user, five queries

[7] https://developers.facebook.com/tools/explorer/.

[8] *HB* stands for hybrid.

are randomly generated from that list. The second set of queries called *qset2* is generated from each user's social data. With *qset2*, we randomly generate 5 queries from users' own Facebook data (e.g., pages they liked, city they live, school they attended). We believe *qset2* is of higher quality since these are the queries that users are very familiar with and hence can understand and evaluate better. (For instance, user living in *Irvine, California* can evaluate the results for query *Irvine, California* very well.). Another benefit of choosing queries from users' Facebook profile is a higher chance of having social actions from the user on the query topic.

As a result, using *qset2* provides us with a better evaluation of our system. Note that in the absence of any social signal, our approaches will perform the same as the baseline approach and hence will not provide many new insights. For the above reasons, we only use *qset1* for the first set of experiments (comparing the main approaches) and use *qset2* for other experiments.

Relevance Assessment. After computing the top-5 results for each of our queries using all approaches, we ran a user study using Amazon Mechanical Turk.[9] One task (hit) was generated for each query. Users of our experiments were typical Mechanical Turk users that were willing to connect using Facebook Connect (and share their data with us) and had at least 50 Facebook friends. We asked workers to log in to our experiment setting using their Facebook account[10] via Facebook connect.[11] For each query and for each worker, top 5 results from all approaches were generated, mixed together (duplicates removed) and presented to the worker. Workers could choose whether each result (Wikipedia article) is *very relevant*, *relevant* or *nonrelevant*. User were not aware of different approaches and could not tell which results is for what approach. Moreover, for each query, we asked each user to provide us with an ordered list of top-5 most relevant documents (from the documents presented) based on his/her own preferences. We use this information to calculate nDCG for each query.

Each task (query assessment) was assessed by 12 workers for query set 1 and 8 workers for query set 2. Each worker was rewarded $0.25 by completion of each assessment.

User Relatedness. To capture the relatedness between two users, we used the total number of interactions between those users (on Facebook) as our metric. We retrieved and counted all the direct interactions (except private messages) between two users and used normalized value of this number as the value of our user relatedness function. Although we could use simpler metrics such as number of mutual friends, we believe that the number of actual interactions is a better representative of relatedness/closeness of two Facebook users than the number of mutual friends between them.[12]

---

[9] mturk.com.

[10] Each volunteer allowed us to read/access his/her Facebook data for this experiment.

[11] https://developers.facebook.com/docs/guides/web/.

[12] For instance, you may have a lot of mutual friends with your high school classmate, without being *close* or *related* to that person. On the other hand, you may not have a lot of mutual friends with your spouse or sister, and still be *close* to them.

Evaluation Metric. We evaluated the accuracy of the methods under comparison using popular nDCG@k and precision@k metrics. nDCG@k and precision@k are the two main metrics used for comparison of different ranking algorithms. Discounted Cumulative Gain (DCG) is a measure for ranking quality and measures the usefulness (gain) of an item based on its relevance and position in the provided list. For comparing different lists with various lengths, normalized Discounted Cumulative Gain (nDCG) is used. It is computed by dividing the DCG by the Ideal Discounted Cumulative Gain or IDCG. The higher the nDCG, the better ranked list. When computing nDCG@k, we considered the ordered list of top-5 results entered by the user as the ideal ordering (IDCG).

Another important metric is precision@k. What matters in many search engines is how many good results there are on the ïňĄrst page or the ïňĄrst three pages (vs. traditional metrics such as recall). This leads to measuring precision at ïňĄxed low levels of retrieved results, such as 10 or 30 documents. This is referred to as precision@k (precision at k), e.g., prec@10. It has the advantage of not requiring any estimate of the size of the set of relevant documents.

The relevance values used for *very relevant*, *somehow relevant* and *not relevant* are 2, 1, and 0, respectively. We calculate prec@k for two scenarios. For the first scenario *prec@k (rel)*, we considered the results evaluated as *somehow relevant* or *very relevant* as relevant. For the second scenario *prec@k* (*vrel*), we only considered the results evaluated as *very relevant* as relevant.

We calculated the final nDCG and precision values by averaging all nDCG and precision values for each query.

4.1 Main Approaches

In the first set of experiments, we evaluate the effectiveness of our three main approaches (rankers) and compare the results with the baseline approach.

The results—prec@5(rel), prec@5(vrel) and nDCG@5—of the four approaches and the two query sets are shown in Tables 1 and 2, respectively. The first observation is that for *qset2*, all our proposed approaches (*TP*, *PT* and *HB*) are noticeably better than the baseline (*BS*) approach. The second observation is that for *qset1*, while *HB* outperform *BS* with regards to all three metrics, the other two social approaches are

Table 1 Main approaches: *qset1*

Approach	prec@5(rel)	prec@5(vrel)	nDCG@5
BS	0.714	0.359	0.760
TP	0.630	0.329	0.652
PT	0.787	0.413	0.655
HB	0.760	0.420	0.815

Table 2 Main approaches: *qset2*

Approach	prec@5(rel)	prec@5(vrel)	nDCG@5
BS	0.787	0.491	0.689
TP	0.856	0.626	0.806
PT	0.890	0.628	0.777
HB	0.846	0.590	0.792

not as successful. This observation plus the first observation show that the hybrid (*HB*) approach is the best approach among all four approaches for all cases. We can also see that while the other two persocial approaches work pretty well for some queries (queries that users already have some related social actions), they may generate less accurate results for random/generic queries (although *PT* still outperforms *BS* for two of the three metrics). This shows that search persocialization works best for queries relevant to the querying user (queries such that the querying user has some social actions on documents relevant to those queries). The third observation is that for both datasets, the margin that our persocial approaches (except *TP* in *qset1*) are better than the baseline approach increases from prec@5(rel) to prec@5(vrel). This shows that if users are looking for *very relevant* results, our proposed approaches generate even better results.

4.2 Persocial Relevance Levels

In this set of experiments, we evaluate and compare the results generated from the three levels of persocial relevance with each other and also with the baseline approach. We use *HB* as our persocial ranker and *qset2* as the dataset. Results for three levels and the *BS* approach are shown in Table 3.

The first observation is that all three levels generate more (or equal for level 1 with regards to nDCG@5(rel) metric) accurate results than the baseline approach in regards to all the three metrics. This not only confirms the fact that our final proposed approach (level 3) generates more accurate results than the baseline, but also shows that even applying one or two levels of our persocial model can improve the search results. The second observation is that in regards to all three metrics, each level improves the accuracy of search results in comparison to the previous level. As we

Table 3 Levels

Approach	prec@5(rel)	prec@5(vrel)	nDCG@5
BS	0.787	0.491	0.689
HB-level1	0.787	0.506	0.730
HB-level2	0.809	0.548	0.744
HB-level3	0.846	0.590	0.792

discussed earlier, each level is built on top of the previous level and complements it by adding mode social signals to the persocial relevance model. in other words, this set of experiments proves our hypothesis and shows that (1) social actions improve the search results, (2) using friends social signals further improves the accuracy of the results, and (3) social expansion also adds to the accuracy of search personalization.

Overall, applying all three levels to the baseline approach will improve both the precision of nDCG of the results significantly. Metrics prec@5(vrel) and prec@5 (vrel) improve from 0.78 and 0.49 to 0.84 and 0.59 (6 and 20 % improvements), respectively. Also, the final ordering of the results in comparison to the ideal ordering (nDCG@5) improves significantly from 0.68 to 0.79 (16 % improvement) as well.

4.3 Friends Versus User

In this set of experiments, we compare the impact of using social data from friends-only, user (querying user) only, or a combination of both on our proposed model. We developed two new variations of *HB* called *HB-UO* (*User-Only*) and *HB-FO* (*Friends-Only*) and compare them with each other and also with the original *HB*. Again, *qset2* is used and social expansion is enabled for all the approaches. Results for the three approaches are shown in Table 4. The first and important observation is that the friends-only approach generates results as effective or even better than those of the user-only approach. This further proves the point that friends' interests and preferences are very similar to the users' own interests and preferences. This finding encourages using friends' actions in the search and ranking process.

The second observation from Table 4 is that *HB* is the best approach among all three (reconfirming the observation that level 2 results are better than level 1 results). As we also saw earlier (for the nonexpanded case), we can see that mixing data from both the querying user and his friends will generate the most accurate results.

4.4 Number of Friends

In this set of experiments, we evaluate the impact of number of friends of the querying user on the accuracy of the results. We categorize users based on their number of friends into three groups: *popular*, *semipopular* and *nonpopular*. *Nonpopular* users are those with fewer that 200 friends (between 50 and 200). *Semipopular* users are

Table 4 User-only versus friends-only

Approach	prec@5(rel)	prec@5(vrel)	nDCG@5
HB	0.846	0.590	0.792
HB-UO	0.823	0.545	0.778
HB-FO	0.831	0.582	0.777

Table 5 Number of friends

Number of friends	prec@5(rel)	prec@5(vrel)	nDCG@5
Popular	0.889	0.626	0.826
Semipopular	0.821	0.564	0.782
Nonpopular	0.780	0.540	0.733

those with fewer than 500 friends and more than 200 friends. Finally, *popular* users are those with more than 500 friends (the most number of friends value among our workers is 1312). We present the prec@5(rel) results for the three groups in Table 5.

The main observation is that the accuracy of the results is directly correlated with the number of friends of the querying user. The *nonpopular* group generates the least accurate results and this is expected since not many social signals from friends and perhaps even from the user himself (users with fewer friends tend to be less active on their social network) are used to influence the search. The *popular* group generates the most accurate results, and *semipopular* group is in between. This observation shows that the larger the amount of data from a user's friends, the persocial relevance scores for that user is more accurate and hence the results generated for that user is improved.

To summarize, the main observations derived from our experimental evaluation are:

- Each level of our persocial model improves the accuracy of search results compared to the previous level. All levels generate more accurate results than the baseline approach.
- For *qset2*, all three proposed approaches generate more precise results and a better ranking than the baseline approach.
- For *qset1*, our proposed *HB* approach, generate more accurate results than the baseline approach (for all three metrics), while the results of the other two approaches vary.
- Results generated only from users' friends social data only is as good (if not better) than the results generated from users' own social actions. The best results are achieved when combining users' own and friends' social data.
- Accuracy of results for each user is directly correlated with the number of friends for that user.

5 Related Work

There are several groups of related studies on the application of social networks in search. With the first group, people through their social networks are identified and contacted directly to answer search queries. In other words, queries are directly sent to individuals and answers to the queries are coming from people themselves [12–14].

In this approach called *search services*, people and their networks are indexed and a search engine has to find the most relevant people to send the queries/questions to. An example of search services is the work in [12]. Except for the work in [12], there are not many academic studies regarding search services. There are also systems based on the synchronous collaboration of users in the search process. HeyStacks [15], as an example of such system, supports explicit/direct collaboration between users during the search. HeyStacks enables users to create *search tasks* and share it with others. HeyStacks is a complementary (and not comprehensive) search engine that needs to work a mainstream search engine to be useful.

In [16, 17], authors show how social platforms (such as Facebook, Linkedin) can be used for crowdsourcing search-related tasks. They propose a new search paradigm that embodies crowds as first class sources for the information seeking process. They present a model-driven approach for the specification of crowdsearch tasks. Crowdsourcing search tasks or *crowdsearching* is a fairly new topic focusing on the active and explicit participation of human beings in the search process. Some interesting models and applications of crowdsearching are presented in [14, 18].

Personalized search has been the topic of many studies in the research community. Search engine can either explicitly ask users for their preferences and interests [19, 20] or more commonly, use data sources related to users' search history such as query logs and click-through data. The most common data source used in search personalization is users' Web (query) log data. Some studies also look at different sources of personal data such as email and desktop files [21]. Recently, few studies started to exploit data from social online systems to infer users' interests and preferences. Xu et al. [22] and Noll and Meinel [23] exploit each user's bookmarks and tags on social bookmarking sites and proposes a framework to utilize such data for personalized search. In a similar paper [24], authors explore users' public social activities from multiple sources such as blogging and social bookmarking to derive users' interests and use those interests to personalize the search.

In [25], authors investigate a personalized social search engine based on users' relations. They study the effectiveness of three types of social networks: familiarity-based, similarity-based and both. In [26], which is a short paper, authors propose two search strategies for performing search on the Web: textual relevance (TR)-based search and social influence (SI)-based search. In the former, the search is first performed according to the classical tf-idf approach and then for each retrieved document the social influence between its publisher and querying user is computed. The final ranking is based on both scores. In the latter, first the social influence of the users to the querying user is calculated and users with high scores are selected. Then, for each document, the final ranking score is determined based on both TR and SI.

In a set of similar papers [27–29], authors propose several social network-based search ranking frameworks. The proposed frameworks consider both document contents and the similarity between a searcher and document owners in a social network. They also propose a new user similarity algorithm (MAS) to calculate user similarity in a social network. In this set of papers, the focus is mainly on user similarity functions and how to improve those algorithms. Majority of their experiments are limited

to a small number of queries on YouTube only. Also their definition of a *relevant* document is somehow ad hoc. A relevant (interesting) result is a result (video) whose category is similar/equal to the dominant category of videos that the searcher has uploaded.

With regards to commercial search engines, Bing and recently Google have started to integrate Facebook and Google+, respectively, into their search process. For some search results, they show the query issuer's friends (from his/her social network) that have *liked* or *+1ed* that result. Their algorithms are not public and it seems that they only show the *likes* and *+1s* and the actual ranking is not affected.

There exists a relevant but somehow different topic of *folksonomies*. Tags and other conceptual structures in social tagging networks are called folksonomies. A folksonomy is usually interpreted as a set of user-tag-resource triplets. Existing work for *social search* on folksonomies is mainly on improving search process over social data (tags and users) gathered from social tagging sites [30–32]. In this context, relationships between a user and a tag and also between two tags are of significant importance. Different ranking models proposed in the context of folksonomies include [33–35]. Studies on folksonomies and/or with focus on social tags/bookmarking face the same limitations of user-based tagging. The main issue with user tagging is that results are unreliable and inconsistent due the lack of control and consistency in user tags [2, 36]. Since there is no standard or limitation on the tags chosen by users are, many problems can arise that lower the quality of the results. As discussed in [36], examples of these issues include: synonymy (multiple tags for the same concept), homonymy (same tag used with different meaning), polysemy (same tag with multiple related meanings), and heterogeneity in interpretations and definitions of terms.

6 Conclusion and Future Work

In this paper, we introduced a novel way for personalization of Web search dubbed persocialized search. With persocialized search, we showed how social actions are relevant and useful to improve the quality of the search results. We proposed a model called persocial relevance model to incorporate three levels of social signals into the search process. In level 1, we showed how to utilize users' own social actions to improve the search results. With level 2, we added social data from users' friends to the proposed model. Finally, in level 3 we proposed social expansion to expand the effect of social action to more documents. Using the persocial relevance model, we proposed three ranking approaches to combine the existing textual relevance models with the persocial relevance models. Furthermore, we developed a system called PERSOSE as a prototype search engine capable of performing persocialized search. Employing PERSOSE, we conducted an extensive set of experiments using real documents from Wikipedia and real user and social properties from Facebook. With several set of experiments, we showed how different levels of our persocial model improve the accuracy of search results. We also evaluated the proposed ranking functions and compared them with each other and a baseline approach.

We believe that in this paper, we defined the overall framework needed for the persocialized search. By design and whenever possible, we allowed for different implementations for the proposed methods. This enables an easier customization as well as optimization of *PERSOSE* for different settings and applications. For any given method, finding the best variations/implemantion for a given context is a general and orthogonal research topic that can and should be pursued by experts of that specific context (e.g., optimal user influence or action weight values should be determined based on a given application and by experts on that application.).

Also, there exist many opportunities to improve and extend the proposed framework in several other directions. Here, we briefly mention several directions of future work or easy extension to apply on our existing framework.

Query Categories. One promising direction to extend persocialized search is to study the effects of persocialization on different categories of queries. We have shown that in general, persocialized search improves the accuracy of the search results. As the next step, it is very reasonable to evaluate this improvement based on different query types and see for what type of queries, persocialized search works best and for what types it works the worst.

Intuitively, persocialized search should work very well with queries that explicitly or implicitly asking for opinions and evaluations. For instance, one can guess that queries on restaurants or movies will benefit significantly when social data from people's friends are integrated into the search process. On the other hand, when users know exactly what they want (e.g., navigational queries), persocialized search will probably have no significant effect.

Studying different query types can also help the system adjust the value of α in Eq. 3 (relative weight of textual and persocialized search relevance) automatically and on-the-fly. In the current system, users are in charge of determining the value of α based on their needs. By calculating value of α based on query categories, this process can be done automatically.

Recommendation and Discovery. With certain domains, the persocialized search described in Sect. 3 can be used to recommend items/documents to the users. For instance, for a online video web site/application, persocialized relevance scores can be used for discovery of interesting videos. When many friends of a user have interacted with a particular video, that video may become of an interest to that user. Recommendation based on our persocial relevance model (level 2) can discover and return such videos.

As another example, imagine a music service integrated with a social network. A song recommendation for such services can possibly benefit using our persocial relevance model. Songs can be suggested to a given user, based on what her friends has listened to or liked while considering friends' influence and closeness to the user.

This type of recommendation is very useful when the textual dimension of the documents do not provide much information about the documents (e.g., empty or few textual terms).

Results Interface. In any type of search, searcher usually needs to know why a returned result is relevant to his search. With textual search, this is often done using document (textual) snippets that contain one or more of the query keywords.

It will be an interesting research problem to study different designs that can add a *social snippet* to a qualified search result. For a given persocialized search result, a simple design would be to add the names of (top k) friends of the querying user who have some social actions on the resulting document plus the actual social actions, underneath the textual snippet. This will take only one extra line while providing to the querying user both the social actions and (close) friend names interacted with the returned document.

References

1. Craig AS, Gregory RG (2005) Connections: using context to enhance file search, 20th ACM symposium on operating systems principles. ACM Press, New York, pp 119–132
2. McDonnell M, Ali S (2011) Social search: a taxonomy of, and a user-centred approach to, social web search. Progr: Electron Libr Inf Syst 45.1, pp 6–28
3. Evans B et al (2009) Exploring the cognitive consequences of social search. In: Proceedings of computer human interaction
4. Vuorikari R et al (2009) Ecology of social search for learning resources. Campus-wide Inf Syst 26(4):272–286
5. Amitay E et al (2009) Social search and discovery using a unified approach. In: Proceedings of hyptertext and hypermedia conference
6. Evans B et al (2008) Towards a model of understanding social search. In: SSM
7. Konstas I et al (2009) On social networks and collaborative recommendation. In: SIGIR
8. Freeman LC et al (1979) Centrality in social networks: conceptual clarification. Soc Netw 1(3):215–239
9. Salton G et al (1987) Term weighting approaches in automatic text retrieval. Technical report. Cornell University
10. Robertson SE et al (1994) Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In: SIGIR
11. Page L et al (1999) The PageRank citation ranking: bringing order to the web
12. Horowitz D et al (2010) The anatomy of a large-scale social search engine. In: WWW
13. Franklin MJ et al (2011) CrowdDB: answering queries with crowdsourcing. In: SIGMOD
14. Yan T, Vikas K, Deepak G (2010) Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones. In: Proceedings of the 8th international conference on mobile systems, applications, and services. ACM
15. Smyth B, Briggs P, Coyle M, O'Mahony MP (2009) Google shared! a case-study in social search. In: User modeling, adaptation and personalization. Springer, pp 283–294
16. Bozzon A, Brambilla M, Ceri S (2012) Answering search queries with crowdsearcher. In: Proceedings of the 21st international conference on World Wide Web. ACM
17. Bozzon A et al (2012) Extending search to crowds: a model-driven approach. Search Comput 7538:207–222
18. Fraternali P et al (2012) CrowdSearch: Crowdsourcing Web search
19. Chirita PA, Nejdl W, Paiu R, Kohlschütter C (2005) Using odp metadata to personalize search. In: SIGIR'05: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval. ACM, New York, pp 178–185
20. Ferragina P, Gulli A (2005) A personalized search engine based on web-snippet hierarchical clustering. In: WWW'05: special interest tracks and posters of the 14th international conference on World Wide Web. ACM, New York, pp 801–810
21. Chirita P-A, Firan CS, Nejdl W (2007) Personalized query expansion for the web. In: 30th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR 2007). ACM, Amsterdam, pp 7–14

22. Xu S, Bao S, Fei B, Su Z, Yu Y (2008) Exploring folksonomy for personalized search. In: SIGIR'08: Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval. ACM, New York, pp 155–162
23. Noll M, Meinel C (2008) Web search personalization via social bookmarking and tagging. The semantic web, pp 367–380
24. Wang Q, Jin H (2010) Exploring online social activities for adaptive search personalization. In: Proceedings of the 19th ACM international conference on information and knowledge management. ACM
25. Carmel D et al (2009) Personalized social search based on the users social network. In: CIKM
26. Yin P et al (2010) On top-k social web search. In: CIKM
27. Gou L et al (2010) SNDocRank: document ranking based on social networks. In: WWW
28. Gou L et al (2010) SNDocRank: a social network-based video search ranking framework. In: MIR
29. Gou L et al (2010) Social network document ranking. In: JCDL
30. Rawashdeh M et al (2011) Folksonomy-boosted social media search and ranking. In: ICMR
31. Schenkel R et al (2008) Efficient top-k querying over social-tagging networks. In: SIGIR
32. Yahia SA et al (2008) Efficient network aware search in collaborative tagging sites. In: VLDB
33. Gulli A, Cataudella S, Foschini L (2009) Tc-socialrank: ranking the social web. Algorithms Models Web-graph 5427:143–154
34. Hotho A, Ja"schke R, Schmitz C, Stumme G (2006) Information retrieval in folksonomies: Search and ranking. In: Sure Y, Domingue J (eds) The semantic web: research and applications, volume 4011 of LNAI. Springer, Heidelberg, pp 411–426
35. Bao S, Xue G, Wu X, Yu Y, Fei B, Su Z (2007) Optimizing web search using social annotations. In: Proceedings of WWW. ACM, pp 501–510
36. Mizzaro S, Vassena L (2011) A social approach to context-aware retrieval. World Wide Web 14(4):377–405

The Pareto Principle Is Everywhere: Finding Informative Sentences for Opinion Summarization Through Leader Detection

Linhong Zhu, Sheng Gao, Sinno Jialin Pan, Haizhou Li, Dingxiong Deng and Cyrus Shahabi

Abstract Most previous works on opinion summarization focus on summarizing sentiment polarity distribution toward different aspects of an entity (e.g., battery life and screen of a mobile phone). However, users' demand may be more beyond this kind of opinion summarization. Besides such coarse-grained summarization on aspects, one may prefer to read detailed but concise text of the opinion data for more information. In this paper, we propose a new framework for opinion summarization. Our goal is to assist users to get helpful opinion suggestions from reviews by only reading a short summary with a few informative sentences, where the quality of summary is evaluated in terms of both aspect coverage and viewpoints preservation. More specifically, we formulate the informative sentence selection problem in opinion summarization as a community leader detection problem, where a *community* consists of a cluster of sentences toward the same aspect of an entity and *leaders* can be considered as the most informative sentences of the corresponding aspect. We develop two effective algorithms to identify communities and leaders. Reviews of six products from Amazon.com are used to verify the effectiveness of our method for opinion summarization.

L. Zhu (✉)
Information Sciences Institute, Los Angeles, USA
e-mail: linhong@isi.edu

S. Gao · S.J. Pan · H. Li
Institute for Infocomm Research, Singapore, Singapore
e-mail: gaosheng@i2r.a-star.edu.sg

S.J. Pan
e-mail: jspan@i2r.a-star.edu.sg

H. Li
e-mail: hli@i2r.a-star.edu.sg

D. Deng · C. Shahabi
University of Southern California, Los Angeles, USA
e-mail: dingxiong.deng@usc.edu

C. Shahabi
e-mail: shahabi@usc.edu

Ö. Ulusoy et al. (eds.), *Recommendation and Search in Social Networks*,
Lecture Notes in Social Networks, DOI 10.1007/978-3-319-14379-8_9

Keywords Product review analysis · Sentiment analysis · Opinion summarization · Social network analysis · Community leader detection

1 Introduction

Nowadays, opinion data can be widely found on the Web, such as product reviews, personal blogs, forums, and news groups. Such information is highly valuable to e-commerce users (e.g., manufacturers, customers, online advertisers, etc.). For example, travelers may rely on comments about hotels on Tripadvisor[1] to book an appropriate resort.

However, the flourish of online opinions is a double-edged sword, which provides useful information meanwhile poses challenges in digesting all the massive information. For instance, in Amazon, some popular products may get hundreds even thousands of reviews, which makes it difficult for potential customers to go through all the reviews to make an informed decision on purchase. Furthermore, some reviews are noninformative and may even mislead customers. To address these issues, most online portals provide two services: aspect summary and review helpfulness rating. Accordingly, various amount of research has been conducted on aspect-based opinion summarization [16, 28, 29, 34, 43, 47] and review quality evaluation [7, 20, 23, 26].

Aspect-based opinion summarization aims to identify aspects of a given entity, and summarize the overall sentiment orientation towards each aspect. For example, for a mobile phone product, aspect-based opinion summarization may return the following information "battery life (three stars); screen (five stars); sound quality (five stars)," where *battery life*, *screen*, and *sound quality* are three of the aspects of a mobile phone, and the numbers of stars denote the corresponding overall sentiment orientation towards the aspects summarized from existing reviews. This kind of summarization is useful for consumers. However, it may lose some detailed information, which is also important for consumers to make decisions. For example, travelers may prefer to get information on suggested traveling routines in detail instead of only summarizing which tourist spots are good or bad.

In some scenarios, opinion summarization by selecting informative reviews is more desirable. Some approaches have been proposed to this task. A common idea behind them is to predict a score for each review to estimate its helpfulness, and select the top ones as informative reviews. However, most of them do not take the following two issues into consideration: (1) **redundancy**, the reviews with highest scores on helpfulness may contain redundant information; (2) **coverage**, the reviews with highest scores on helpfulness may not cover all aspects of the entity, and some important aspects may be missing.

In our prior work [46], we have proposed a new opinion summarization framework, named *sentence-based opinion summarization*, to address these issues. Given a

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

set of reviews for a specific entity, the goal of sentence-based opinion summarization is to extract a small subset of informative sentences to represent the reviews, under the assumption that important sentences are origins of topics and opinions. Importance analysis is widely studied in various areas such as business management, social network analysis, and so on. In the early 1900s, economists have observed the Pareto principle [5]: where something is shared among a sufficiently large set of participants, there must be a number k between 50 and 100 such that "k % is taken by $(100 - k)$ % of the participants." In the same way, in this work, given a piece of opinion text, we focus on extracting a small number of sentences that cover the great mass of opinions and topics and generating a summary for it. The quality of summary is evaluated in terms of the coverage of the entity aspects and the polarity distribution preservation of the aspects (i.e., positive, negative, or neutral). In other words, we aim to generate summaries by extracting a small number of sentences from the reviews of a specific entity, such that the coverage of the entity aspects and the polarity distribution of the aspects can be preserved as much as possible. Note that the proposed framework is not to resume aspect-based opinion summarization approaches. In contrast, since the selected informative sentences preserve the coverage and sentiment polarity distribution of the entity aspects, aspect-based opinion summarization techniques can be post-applied to the selected sentences to generate summarization towards each aspect without information loss. Figure 1 depicts the relationship between our sentence-based opinion summarization and aspect-based opinion summarization.

Based on our opinion summarization framework, we propose a graph-based method to identify informative sentences. More specifically, we formulate the informative sentence selection problem in opinion summarization as a community and leader detection problem in social computing. A sentence graph is constructed by adding an edge between a pair of sentences if they are similar in both word distribution and sentiment polarity distribution. Subsequently, each node of the graph representing a sentence can be considered as a user in social computing. Thus, in the sentence graph, a community consists of a set of sentences towards the same aspect and the leaders of the community can be considered as the most informative sentences.

Finally, we propose two algorithms to detect leaders on the sentence graph. We first propose a *C*lique-based *C*ommunity and *L*eader detection algorithm (CCL), where we find overlapping communities by enumerating all maximal cliques and then model the community leader detection problem as a budgeted maximum coverage problem. The CCL algorithm is able to well preserve both aspect coverage and polarity distribution. However, there are some limitations of CCL in terms of

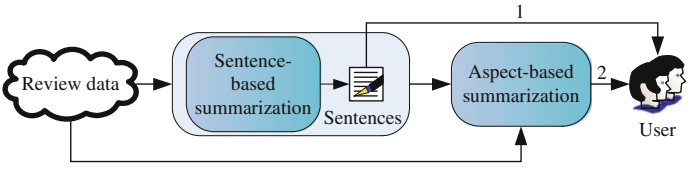

Fig. 1 Sentence-based summarization versus aspect-based summarization

efficiency (enumerating all maximal cliques is very time-consuming) and parameter-free issue (the size of summary is highly dependent on the parameter setting). To this end, we further develop an alternative algorithm, which aims to *S*imultaneously detect *C*ommunities and *L*eaders on the sentence graph (SCL), where communities are formed by assigning other sentences to leaders (i.e., informative sentences), and leaders are selected according to their informativeness in both documents and communities. Though SCL obtains lower aspect coverage than CCL, it is a good trade-off between efficiency and effectiveness. In addition, our user study shows that SCL is preferred by real users in terms of conciseness. Therefore, if aspect-based opinion summarization is required to be post-applied to the selected sentences (Case 2 in Fig. 1), CCL would be a better choice with less information loss. When a summary which is generated from selected sentences is directly displayed to users (Case 1 in Fig. 1), we suggest using the SCL algorithm with conciseness.

In all, we summarize our contributions of this research:

- We have introduced a new sentence-based summarization framework which generates summaries that preserve aspect coverage as much as possible and are representatives of aspect-level viewpoints.
- We have bridged across the area of sentiment analysis and the area of social computing by applying community and leader detection algorithm to solve the informative sentences selection problem.
- We have presented two effective leader community detection algorithms, namely clique-based community and leader detection algorithm (CCL) and simultaneous community and leader detection algorithm (SCL), to find informative sentences from a sentence graph.
- We have conducted experiments using real data collected from Amazon product review and two evaluation metrics "aspect coverage" and "polarity distribution preservation." Our experimental results demonstrated the effectiveness of the proposed technique.

2 Related Work

The most related work to ours is sentiment summarization [3, 11, 21] where a summary is built by extracting representative bits of text from a set of documents. Lerman et al. [21] aim to generate summaries that are representative of the average opinion and cover important aspects when aspect set is given. The quality of summary is evaluated in terms of the mismatch between the average sentiment of a summary and the known sentiment of an entity and the coverage of aspects. The goal of our work is more fine-grained: to generate summaries that maximize aspect coverage and preserve the aspect-level viewpoints of an entity without knowing aspect set in advance. Another work which is known as Opinosis [11], aims to generate concise abstractive summaries of highly redundant opinion data for a specific aspect (i.e., battery life for kindle). The key idea of Opinosis is to use a word graph to represent the opinion data, and then repeatedly find paths through the graph to produce concise

summaries. However, our work differs in both the problem setting and methodology from Opinosis. In our work, aspects are unknown and sentences are not grouped via aspects in advance, while Opinosis takes groups of sentences towards different aspects as inputs. In addition, our method uses sentence graph and detects leaders of sentence community to generate concise summaries instead of using word graph and finding paths to generate summaries.

Besides sentences/words selection, aspect-based approaches are another important branch of sentiment summarization, which includes three distinct steps: aspect/feature identification [16], sentiment prediction/classification [12, 36, 44, 45], and summary generation [28]. According to a survey on opinion summarization [19], most of the existing works use three kinds of approaches to perform aspect identification: mining techniques [15, 16], Natural Language Processing (NLP)-based techniques [34], and integrated techniques [4, 14, 28, 29, 39, 47]. In this work, we propose a new sentence-based summarization framework, whose objective is totally different from those aspect-based summarization approaches.

Review quality prediction is another branch of related works [7, 20, 22, 23, 26], which aims to estimate scores for each review, and rank the reviews based on the scores. Recently, Tsaparas et al. [40] propose to select a comprehensive subset of reviews to represent the original reviews. In their work, the review selection problem is modeled as a maximum coverage problem and several heuristic algorithms are proposed to greedily select a set of comprehensive reviews. Our work is different from theirs in two ways: (1) our opinion data selection is done in the sentence level rather than the review level, and (2) we model the summarization problem as a community leader detection problem in sentence graph.

Our work is also related (but not highly relevant) to existing works on multidocument summarization via sentence selection [8, 17, 25, 30, 31, 41], subjective summarization [33], and sentence compression for single-document summarization [9, 42]. In document summarization, the objective is to summarize the information content in the document with shorter texts, while the opinion summarization task focused on features or objects on which customers have opinions. In addition, our methodology differs from previous graph-based ranking methods such as textrank [30, 31] and clustering-based techniques such as [41] on multidocument summarization. Compared with textrank, our work generate summaries using both the sentence-sentence term similarity and the sentiment polarity information; while in textrank, either the sentiment polarity information of the sentences or the intersection between sentences is not taken into consideration. Though community detection is essentially a clustering problem, we highlight that our method differs from previous clustering-based techniques in the following aspects: (1) Our goal is to select informative sentences rather than group similar sentences together, thus our main focus is to detect leaders. (2) The extracted leaders are different from the centroids of clusters. A centroid represents a statistical high relevance to a cluster of sentences, but may suffer from the low informative and manipulated issues. For instance, a cluster of sentences may consist of all spam reviews resulting in the centroid sentence being of low quality. Instead, our leader detection algorithm makes use of informativeness within a community and within a review to select high-quality sentences.

3 Problem Formulation

Denote x a specific entity that consists of a set of aspects $\mathcal{A} = \{ a_1, a_2, \ldots, a_m \}$, and a set of reviews on the entity $\mathcal{R} = \{D_1, D_2, \ldots, D_l\}$, where D_i ($i = 1$ to l) represents a review. Each review D_i consists of several sentences $D_i = \{s_1, s_2, \ldots, s_{n_i}\}$, where s_j ($j = 1$ to n_i) represents a sentence. Define $|D_i| = n_i$ the size of the review D_i, and $|\mathcal{R}| = \sum_{i=1}^{l} |D_i|$ the size of the review set $\mathcal{R}$.

Based on the above terminologies, the informative sentence selection problem is defined as follows:

Problem 1 (*Sentence-based opinion summarization*) Given a set of reviews $\mathcal{R}$ on a specific entity x, which consists of a set of aspects $\mathcal{A}$, our goal is to find a few number of sentences $\mathcal{S}$ where $|\mathcal{S}| \ll |\mathcal{R}|$ such that $\mathcal{S}$ covers the aspects in $\mathcal{A}$ as many as possible and preserves the **aspect-level** sentiment polarity distribution of $\mathcal{R}$ as much as possible. Note that both the aspect set $\mathcal{A}$ and their sentiments are unknown in training.

The goal of Problem 1 is to generate a summary of documents that is representative of the average aspect-level sentiment. We provide more perspectives of Problem 1 using the following example.

Example 1 Figure 2 shows an example of six sentences from four reviews discussing about an entity ipad protector. Though both aspects and sentiments are unknown, here we also list them in the right Table to help illustrate what a good solution is to Problem 1. From the example, the overall sentiments of review D_1, D_2, D_3 and D_4 are positive (+), positive (+), negative (−), negative (−), respectively. The average sentiment for aspect "price" is positive (+/−: 2/1) and is negative (+/−: 1/3) for aspect "bubble"; while the overall sentiment toward ipad protector is neutral (+/−: 2/2). A possible solution to the Problem 1 is the summary $\{s_4, s_5\}$, which looks good since it covers both aspects and preserves the overall neutral sentiment. However, this summary is misleading, especially to users who are concerned about aspect "price" (Most of the reviewers feel price is good, while this summary states a very negative

1. D_1 Definitely worth the price!
2. D_2 For the price, I would recommend it.
3. D_1 They get bubbles, and a pain to apply.
4. D_2 Completely bubble-free, good protector!
5. D_3 Lured by the low price tag for these iPad protectors, I tried to apply them twice and finally gave up
6. D_4 No matter what I did I could never get the air bubbles out and ended up wasting 2 out of 3 covers.

sentence	aspect-level sentiment	review-level sentiment
s_1	price +	D_1 +
s_2	price +	D_2 +
s_3	bubble -	D_1 +
s_4	bubble +	D_2 +
s_5	price -, bubble -	D_3 -
s_6	bubble -	D_4 -

Fig. 2 An illustrative example

opinion toward price). Instead, the summary $\{s_1, s_5\}$ that preserves the aspect-level sentiment is more meaningful and a better solution to Problem 1.

It is nontrivial to deal with Problem 1. One may formulate it as an optimization problem $\arg\max_{\mathcal{S}\subseteq\mathcal{R}} f(\mathcal{S})$ where f denotes a scoring function over possible summaries. The definition of f can take the aspect coverage and aspect-level sentiment difference between summary $\mathcal{S}$ and review set $\mathcal{R}$ into consideration. However, since both aspect set and aspect-level sentiment are unknown, it is difficult to estimate either the aspect coverage or the sentiment difference, not to mention to embed them into f. Besides, even solving f is possible, usually tackling optimization problem is typically NP-hard.

Another method to solve Problem 1 is to group sentences toward similar aspects into a cluster, and select representative sentences from each group to generate summaries.

Generally, our solution is a combination of these two methods. An overview of our proposed framework is summarized in Fig. 3, where we also group sentences into communities and extract informative sentences from communities by solving an optimization problem. Instead of using the content information (e.g., term vectors) to group sentences, we utilize the term similarity and sentiment polarity distributions to build graphs and then group sentences based on structure proximity. Since the sentence graph is built on texts, it identifies connections between various sentences in a corpus, and implements the concept of recommendation. The nodes that are highly recommended by other nodes in the sentence graph are likely to be more informative for the given corpus. Therefore, with the sentence graph, the informative sentence selection problem can be formulated as a leader identification problem in the sentence graph. We then propose two algorithms to detect communities (a group of sentences S_i which are related to a specific aspect a_i and have similar sentiment polarity distributions toward a_i) and leaders (informative sentences). After that, a set of informative sentences are extracted from each community and a system summary is generated accordingly. We discuss the details in the next section.

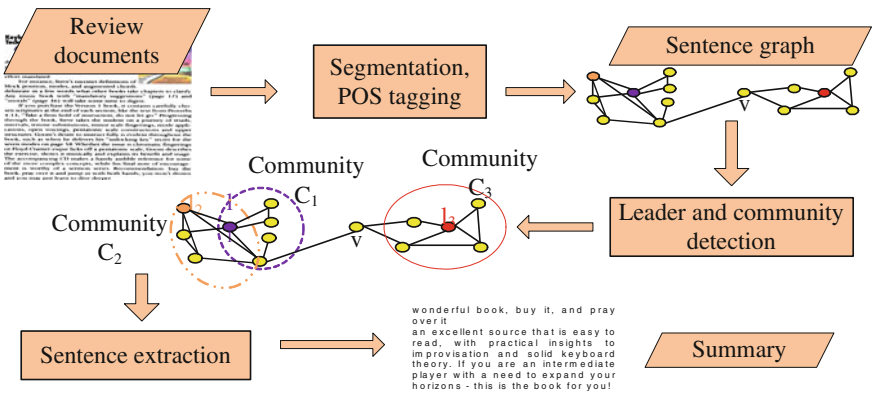

Fig. 3 An overview of the proposed opinion summarization framework

4 Methodologies

In this section, we first provide the details of sentence graph computation in Sect. 4.1. Then in Sect. 4.2, we introduce a clique-based community and leader detection algorithm, where each maximal clique represents a community and leaders are detected by solving budgeted maximum coverage problem. Next in Sect. 4.3, we propose an algorithm which simultaneously identify both communities and leaders. Finally, we summarize our opinion summarization framework in Sect. 4.4.

4.1 Sentence Graph Construction

Denote $G = (V, E)$ the sentence graph constructed from the set of sentences $S = \{s_1, s_2, \ldots, s_n\}$, where each node $v \in V$ represents a sentence and each weighted edge $e \in E$ evaluates the similarity between the two corresponding sentences. A key research issue in sentence graph construction is to design a function to measure similarity between sentences. Before presenting the similarity function we used in this paper, we first introduce two definitions.

Definition 1 (*Term Similarity*) Given two sentences s_i and s_j, their term similarity is defined as

$$\tau(s_i, s_j) = \cos(\overrightarrow{v_i}, \overrightarrow{v_j}),$$

where $\overrightarrow{v_i}$ and $\overrightarrow{v_j}$ are the term vector representations of s_i and s_j, respectively, and $\cos(\cdot)$ denotes the cosine similarity function.

Definition 2 (*Adjective Orientation Similarity*) The adjective orientation similarity of two sentences s_i and s_j is defined by the following equation:

$$\alpha(s_i, s_j) = 1 - \frac{|\sum_{t_i \in s_i} SO(t_i) - \sum_{t_j \in s_j} SO(t_j)|}{|\sum_{t_i \in s_i} SO(t_i) + \sum_{t_j \in s_j} SO(t_j)|},$$

where $t_i \in s_i$ (or $t_j \in s_j$) denotes an adjective term in sentence s_i (or s_j), and $SO(t_i)$ (or $SO(t_j)$) denotes the probability of t_i (or t_j) being positive, which is derived from the Semantic Orientation Dictionary [37].

As mentioned in the previous section, we aim to group the sentences that are toward the same aspect and have similar sentiment polarity orientation into a community. Therefore, the above two similarities are both important for constructing the sentence graph. As a result, we define our similarity function between sentences as follows:

$$\mathtt{sim}(s_i, s_j) = \lambda\tau(s_i, s_j) + (1 - \lambda)\alpha(s_i, s_j), \tag{1}$$

where $\lambda \in [0, 1]$ is a trade-off parameter to control the contribution balance between the term and adjective orientation similarities.

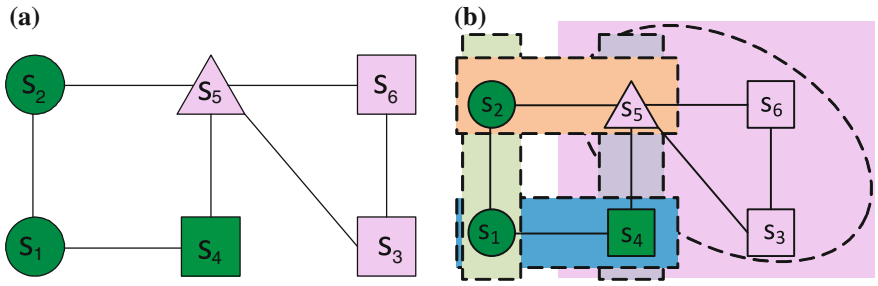

Fig. 4 **a** shows the sentence graph of Fig. 2. *Circles* represent sentences with aspect "price," *squares* represent sentences with aspect "bubble," and *triangles* represent sentences with both aspects. *Green nodes* denote sentences with positive sentiments and *pink nodes* denote sentences with negative sentiments. A set of overlapping communities, $\mathcal{C}=\{\{s_3, s_5, s_6\}, \{s_1, s_2\}, \{s_1, s_4\}, \{s_2, s_5\}, \{s_4, s_5\}\}$, which is computed based on maximal cliques, is shown in (**b**)

Given the similarity function, we link sentences s_i and s_j with an edge associated with an nonnegative weight w_{ij} as follows:

$$w_{ij} = \begin{cases} \texttt{sim}(s_i, s_j), & \text{if } s_i \in \mathcal{N}_k(s_j) \text{ or } s_j \in \mathcal{N}_k(s_i), \\ 0, & \text{otherwise,} \end{cases}$$

where $\mathcal{N}_k(s_j)$ is the k-nearest neighbors of the sentence s_j according to the similarity measure.[2] From the preliminary test, we use a grid search to find the best combination for λ and k. The optimal values we found are $\lambda = \frac{2}{3}$ and $k = \lceil \frac{N}{5} \rceil$, where $N = |\mathcal{R}|$. Therefore, for all the experiments in this paper, we set $\lambda = \frac{2}{3}$ and $k = \lceil \frac{N}{5} \rceil$.

Example 2 Figure 2 shows an example of four review documents for ipad protector with six sentences in total. The associated sentence graph with $k = 2$ is constructed in Fig. 4a. Each node is linked with its k-nearest neighbors as well as reversed k-nearest neighbors. For clearer representation, the figure only shows the edges but the weights are omitted. Note that not every node has the same degree k since a node can be reversed k-nearest neighbors of many nodes.

4.2 Clique-Based Community and Leader Detection Algorithm (CCL)

Intuitively, since edges in sentence graph are created based on the similarity of sentences, we can make the assumption that a group of highly connected sentences

[2] Note that $|\mathcal{N}_k(s)|$ can be larger than k since there could be the event of ties (i.e., a set of neighbors have the same similarity to s).

are more likely to share the same topic. Therefore, we find the set of all maximal cliques in sentence graph and each maximal clique forms a community.

More specifically, given a graph G, a clique in G is a subset of vertices, $C \subseteq V$, such that the induced subgraph by C is a complete graph in G. C is called a maximal clique (maxclique in short) in G if there exists no clique C' in G such that $C' \supset C$. For example, consider the sentence graph shown in Fig. 4a, the set of all maximal cliques are $\{s_3, s_5, s_6\}$, $\{s_1, s_2\}$, $\{s_1, s_4\}$, $\{s_2, s_5\}$, and $\{s_4, s_5\}$. Therefore, we can compute a set of overlapping communities based on maximal cliques, as shown in Fig. 4b.

In the development of this system, we adopt an efficient algorithm proposed by Cheng et al. [6] to find the set of all maximal cliques.

Once we have a set of overlapping communities, the next focus is to identify a set of leaders (i.e., informative sentences) to generate a concise summary. Recall in Sect. 1, we have raised two critical issues for a concise summary: redundancy and converge. Thus, we investigate two principles that a set of leaders should have: good aspect coverage and informativeness. Aspect coverage accesses whether the set of selected leaders $\mathcal{S}$ have well captured all the communities representing subtopics, while informativeness evaluates whether the selected leaders well represent the communities they belong to. In addition, since users demand a concise summary, the size of a summary (i.e., total number of words) cannot exceed a given budget.

Intuitively, if we know the informativeness of each node (e.g., relative importance score) in the community, we then start picking up high informative nodes from each community until all the communities are covered or the size of summary reaches the budget. This discussion motivates us to formulate the leader detection problem as a budgeted maximum cover problem [1] as follows:

Problem 2 (*Leader Detection Problem*) Given a sentence graph $G = (V, E)$ where each sentence s is associated with a penalty cost $w(s)$ and a informativeness score $\varphi(s)$, its overlapping communities $P = \{C_1, C_2, \ldots, C_m\}$ where each group of sentences C_i ($i = 1$ to m) represents a subtopic, and a number $\mathcal{B}$, the leader detection problem is to find a subset of sentences $\mathcal{S} \subseteq V$ such that the cost of $\mathcal{S}$ is within budget ($w(\mathcal{S}) \leq \mathcal{B}$) and the reward of covering communities (which is denoted as $\varphi(P \cap \mathcal{S})$) is maximized.

Naturally, in this work, the penalty cost $w(s)$ of each sentence s is defined as the total number of words in the sentence s. Regarding the informativeness score $\varphi(s)$, as we know, the centrality of nodes in a community measures the relative importance of nodes within the group. Therefore, we consider a sentence to be informative if it has high centrality within its community in the sentence graph. There are many measures of centrality that could be parameters to the algorithm, namely degree, betweenness [10], closeness [35], and eigenvector centrality measures [32]. We experimented with all of them and based on our results, we selected the degree centrality for the default measure which yields the most accurate results in most of the cases and also is easy to compute.

The degree centrality of the node v within the community C is simply the number of edges from the community incident upon the node v and represents to some extent

the "popularity" of v in the community. That is,

$$\texttt{deg}(v, C) = \frac{\sum_{u \in C} w(u, v)}{|C| - 1}$$

where each edge (u, v) denotes an edge in C that is incident to node v, and w is the weight of the edge.

Since a sentence may be inside more than one community of P in the sentence graph G, we then further define $\varphi(s)$ as

$$\varphi(s) = \frac{1}{|\mathcal{C}_s|} \sum_{C \in \mathcal{C}_s} deg(s, C) \tag{2}$$

where $\mathcal{C}_s$ denotes a set of communities which contain sentence s.

We have discussed the details of penalty cost $w(s)$ and informativeness of sentence $\varphi(s)$, now let us turn our attention to the solution of Problem 2. Unfortunately, the budgeted maximum cover problem is known to be NP-hard for general graphs and approximation algorithms are needed [1, 18]. Hence, we develop a greedy algorithm, which iteratively adds an important but cheap node, to solve Problem 2. The details are shown in Algorithm 1.

Start with an empty sentence set $\mathcal{S}$ (line 1), in each iteration, this greedy algorithm picks up a sentence s^* from those uncovered partitions which maximizes the marginal gain (lines 3–4). Furthermore, after s^* is added to $\mathcal{S}$, it is required to update the set of covered communities: all the communities that contain s^* can be marked as covered (lines 5–7). The algorithm stops and returns $\mathcal{S}$ until the budget is exhausted or all the communities are covered (lines 8–9).

Bounds of the Greedy Algorithm. Khuller et al. [18] have proved that for nondecreasing reward φ and nonnegative penalty cost w, there exists a greedy algorithm with an approximation factor of $\frac{1}{2}(1 - \frac{1}{e})$. Note that in Algorithm 1, φ is nondecreasing and w is nonnegative, hence following the proof in [18], we show that Algorithm 1 achieves an approximation factor of $\frac{1}{2}(1 - \frac{1}{e})$ as well. And the worst case running

Algorithm 1 Clique-based Leader Detection $\texttt{CCL}(V, P, w, \varphi)$

Input: sentence set V, clique-based communities P, cost function w and φ, budget $\mathcal{B}$
Output: a subset sentences $\mathcal{S}$
1: $\mathcal{S}$=∅, X'=∅
2: **repeat**
3: s^*=arg $\max_{v \in (V \setminus X')} \{\frac{\varphi((\mathcal{S} \cup v) \cap P) - \varphi(\mathcal{S} \cap P)}{w(v)}\}$ subject to $w(\mathcal{S} \cup \{v\}) \leq \mathcal{B}$
4: $\mathcal{S}$=$\mathcal{S} \cup \{s^*\}$
5: **for** each $V_i \in P$
6: **if** $s^* \in V_i$
7: X'=$X' \cup V_i$
8: **until** $w(\mathcal{S}) \geq \mathcal{B}$ or $X' == V$
9: **return** $\mathcal{S}$

time of this algorithm is bounded by $O(\mathcal{B}|V|)$ where $|V| = |\mathcal{R}|$ denotes total number of sentences in review set.

4.3 Simultaneous Community and Leader Detection Algorithm (SCL)

In the previous section, we have proposed a sequential algorithm to first identify communities by enumerating all maximal cliques and then detect leaders by solving the budgeted maximum coverage problem. However, there are some limitations of the proposed CCL algorithm: first, the size of leader set (i.e., summary) highly depend on the parameter $\mathcal{B}$. Next, in the CCL Algorithm, the leader sentences are selected according to the only two criteria aspect coverage and representativeness. In real application such as Amazon, each review may have a helpful vote number which indicates the quality of review itself. We suggest that the quality of review is helpful for identifying informative sentences, with the assumption that a sentence from a more helpful review is more informative than another low-quality review.

To address the aforementioned issues, we propose an alternative leader detection algorithm, namely simultaneous community and leader detection algorithm (SCL). The general idea is similar to the k-mean clustering: we first initialize a set of leaders with high degrees and then assign other nodes to each leader to form communities. Given each community, we then update its leaders based on the informativeness of sentences within both communities and reviews. We iteratively repeat the above process until there is no change in leadership. An overview of SCL algorithm is outlined in Algorithm 2.

There are several advantages of the proposed SCL algorithm in terms of parameter-free property and efficiency. First, the number of leaders and the size of summary is automatically determined by Algorithm 2. There is no requirement for additional parameters such as $\mathcal{B}$ in the CCL algorithm. Next, instead of first using the very time-consuming maximal clique enumeration approach to find communities and

Algorithm 2 Simultaneous Community and Leader Detection $\texttt{SCL}(G)$

Input: a sentence graph G and review score $\varphi(D)$ for each review D
Output: a subset of sentences $\mathcal{S}$
1: Initialize a set of leader $\mathcal{S}$ in G: $\mathcal{S}$=$\texttt{LL}(G)$ (Algorithm 3)
2: Initialize communities $\mathcal{C}$ by assigning each leader to a single community
3: **repeat**
4: let followers F=$\{v \in V | v \notin \mathcal{S}\}$ and order $v \in F$ by its distance to $\mathcal{S}$
5: **for** each $v \in F$
6: update community set: $\mathcal{C}$=$\texttt{Community}(G, \mathcal{S}, v, \mathcal{C})$ (Algorithm 4)
7: **for** each sentence $s \in \mathcal{S}$ with $s \in D$
8: update community leader for $C(s)$ and review score $\varphi(D)$ (Eq. 3)
9: update $\mathcal{S}$ as new leaders of each community of $\mathcal{C}$ (Eq. 4)
10: **until** no change in the leader list $\mathcal{S}$
11: **return** $\mathcal{S}$

then another approach to detect leaders in the CCL algorithm, the SCL algorithm efficiently determines communities and leaders simultaneously in a unified framework. Finally, in the SCL algorithm, the leader sentences are selected based on not only their informativeness in communities but also qualities of reviews they belong to.

We now present more details about each important step in the Algorithm 2: Leader initialization, Community assignment, and Leader reassignment.

4.3.1 Leader Initialization

Once the sentence graph is built, we can initialize some nodes of the graph as leaders and iteratively identify and update the communities and leaders. The naïve initialization is to randomly select k sentences from the sentence graph as leaders. This is simple to implement, but is not deterministic and may produce unexpected results. Another approach is to select a set of global top sentences such as selecting k sentences that have highest degrees in the sentence graph. However, choosing arbitrarily top-k high-degree sentences may suffer from the redundancy and low coverage issues. An extreme case is that all of the top-k sentences are discussing about the same aspect and hence the results are not satisfied.

As an alternative, we want to select a set of leader sentences that are well distributed in the sentence graph (i.e., to avoid choosing leaders from the same community). More specifically, a node v in the sentence graph is selected as an initial leader if

1. It is a h-node in sentence graph G, and
2. None of its neighbors is a leader.

The key component of our lead initialization is the largest set of h nodes in sentence graph G that have degree at least h, called the h-node [6]. The concept of h-node is originated from the h-index [13] that attempts to measure both the productivity and impact of the published work of a scientist or scholar. Putting it into the concept of our sentence graph, a h-node in sentence graph corresponds to a sentence that is similar to at least another h sentences and to a certain extent represents the "ground truth." Therefore, it is straightforward to adopt the h-node concept for initial leadership evaluation. Note that the h value and the set of h-nodes can be computed easily using a deterministic and parameter-free algorithm proposed by [6].

Another component of our leader initialization aims to reduce redundance and achieve better community coverage. After finding the set of h-nodes, we start from the node with highest degree, and add the next higher degree h-node to the current set of leaders if it is not a neighbor of any of the already selected leaders. All the details of the leader initialization are outlined in Algorithm 3.

4.3.2 Community Assignment

Once some leaders are initialized, we can initialize communities by assigning each leader to a single community. After that the community membership of the remaining nodes can be determined by assigning them to nearby leaders. The intuitive idea is

Algorithm 3 Leader Initialization LL(G)

Input:Graph G=(V, E)
Output: a set of leaders L
1: L=∅
2: Compute the set of h-nodes $H \in V$ and sort H by node degree in descending order
3: **while** H is not empty
4: pick v from the front of H
5: H=$H \setminus \{v\}$, flag=true;
6: **for** each $s \in L$
7: **if** v is a neighbor of s
8: flag=false;
9: **if** flag==true
10: L=$L \cup \{v\}$
11: return L

similar to label propagation algorithms for link-based classification [27, 38], where class labels (i.e., community membership in our scenario) of linked nodes are correlated. Therefore, a node is assigned to a community if most of its neighbors have already resided in the community.

Algorithm 4 presents the method to determine the community membership for a node v. Note that in Algorithm 2 (Lines 6–7), we start calling Algorithm 4 for nonleader nodes with ascending order of distances to leaders. By doing this, we iteratively propagate the community membership from leaders to royal members (i.e., neighbors of leaders), and then to the descendants of royal members (i.e., n-hop neighbors of leaders).

Example 3 Figure 5 shows an example of sentence graph with two communities formed by leader l_1 and l_2. Assume that each edge is equally weighted, then node v should be assigned to leader l_1 since v shares more common neighbors with community C_1 than C_2. Consider another extreme case where edges connecting v and

Algorithm 4 Update community Community(G, L, v, $\mathcal{C}$)

Input: graph G, leaders L, node v, communities $\mathcal{C}$
Output: A refined community set $\mathcal{C}$
1: let max = 0, l_t=-1
2: **for** each $l \in L$
3: let N denote the set of common neighbors between v and community $C(l)$
4: δ=$\sum_{u \in N} w(u, l) + w(u, v)$
4: **if** δ >max
5: max=δ
6: l_t=l
7: **if** l_t is negative
8: mark v as outlier
9: **else**
10: update $C(l_t) \in \mathcal{C}$ to $C(l_t) \cup \{v\}$
11: **return** $\mathcal{C}$

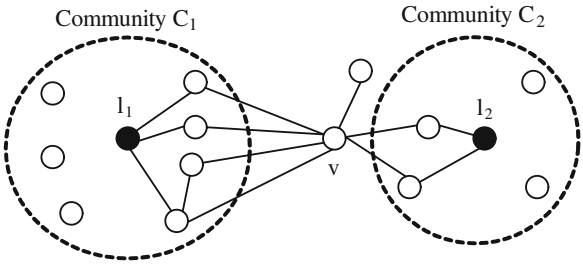

Fig. 5 An example of community membership determination

nodes in C_1 are with weight 0.001 and edges connecting v and nodes in C_2 are with weight 0.9, then node v is assigned to leader l_2 since it is more similar to community C_2 in terms of content and polarity similarity.

4.3.3 Leader Reassignment

As we have discussed earlier, in CCL algorithm, the informativeness of a sentence is only evaluated by its degree centrality within the community. However, we argue that the informativeness of a sentence is related to not only its representative within the community, but also the quality of the review it belongs to. More specifically, we have the following two assumptions:

1. A review is important if it contains lots of informative sentences;
2. A sentence is informative if it appears in an important review.

Hence, given a sentence s from a review D, which is represented as a node v in the sentence graph and is in the community $C(s)$, the informativeness of the sentence $\varphi(s)$ is defined as follows:

$$\begin{cases} \varphi(s) = \ \varphi(D)\mathtt{deg}(v, C(s)), \\ \varphi(D) = \frac{1}{|D|}\sum_{s\in D}\varphi(s), \end{cases} \tag{3}$$

where $\mathtt{deg}(v, C(s))$ is the degree centrality of the node v within the community $C(s)$, and $\varphi(D)$ denotes the importance of a review D. Without any prior knowledge, for each review $D \in R$, we can just initialize the $\varphi(D)$=$1/l$ where l is number of reviews. However, when additional information such as "helpfulness" rating score of each review is known in advance, we can initialize the value of $\varphi(D)$ as the "helpfulness" score.

Based on Eq. 3, we can update the $\varphi(s)$ and $\varphi(D)$ mutually in each iteration. After that, for each community, the sentence with the highest informativeness score is selected as the new leader,

$$s^* = \arg\max_{s\in C(s)} \varphi(s) \tag{4}$$

4.4 Summary Generation

We now conclude the proposed sentence-based opinion summarization using the following example:

Example 4 Given a set of reviews shown in Fig. 2, in our sentence-based opinion summarization, the first step is to construct a sentence graph shown in Fig. 4a. Next, we either use the CCL algorithm to find a set of clique-based communities, as shown in Fig. 4b and then the sentences s_1, s_5 are extracted for summary. Or as an alternative, we use the SCL algorithm to find communities $\{\{s_1, s_2, s_4\},\{s_5, s_3, s_6\}\}$ and leaders $\{s_1, s_5\}$ simultaneously. Both of the two algorithms result in a summary with two sentences and about 24 words. A manually generated Aspect-based summary, which can be considered as a reference summary, is "Price: 4 stars and Bubble: 1.5 stars". We observe that our summary does not lose any aspect coverage. In addition, there is no mismatch of sentiments for any aspect between our system summary and manual summary. Therefore, from the comparison, we can conclude that our Leader-based summary covers as many aspects as manual summary and selects most of the informative sentences. What's more, it is more convenient to generate the manual Aspect-based summary from our system summary than from the original reviews in Fig. 2.

5 Experiments

5.1 Datasets

The dataset[3] used in our experiments is a collection of product reviews crawled from Amazon.com. The reviews are about six product domains: *Belkin case* (case), *Dell laptop* (laptop), *Apple iMac* (iMac), *Apple ipad* (ipad), *ipad protector* (protector), and *Kindle* (kindle). Each review contains a title, review content, reviewer information, and an overall rating. The labeling of polarity of each review mainly depends on the given overall rating. In addition, for each product domain, we manually label its aspects and the sentiment polarity towards them on each sentence. The detailed information of the dataset is summarized in Table 1.

5.2 Evaluation Metric

We evaluate the proposed method together with the three baselines using two metrics: the aspect coverage and the polarity distribution preservation.

Aspect coverage: Given the review set $\mathcal{R}$ with a set of aspects $\mathcal{A}$, the aspect coverage of a summary $\mathcal{S}$ is defined as

[3] Available at http://sites.google.com/site/linhongi2r/data-and-code.

Table 1 Summary of the dataset

Product	No. of reviews	No. of sentences $\lvert\mathcal{R}\rvert$	Percentage of positive reviews (%)	No. of aspects $\lvert\mathcal{A}\rvert$
Case	625	2,865	83	19
Laptop	68	469	39.2	23
iMac	34	567	74	17
ipad	218	3,572	61	41
Protector	141	953	64	20
Kindle	1,858	21,948	72.6	43

$$\zeta = \frac{|\{a_i | a_i \in \mathcal{A}, a_i \in \mathcal{S}\}|}{|\mathcal{A}|} \times 100\,\%$$

Note that higher value of ζ implies better aspect coverage.

Polarity distribution preservation: Given the review set $\mathcal{R}$ and the aspect set $\mathcal{A}$, the aspect-level polarity distribution of $\mathcal{R}$ can be represented as a vector $\overrightarrow{t} = (t_1, \ldots, t_n)$ with length $3 \times |\mathcal{A}|$ where t_{3i-2}, t_{3i-1} and t_{3i} denote the percentage of positive, negative, and neutral sentences that are related to aspect a_i ($i = 1$ to $|\mathcal{A}|$) respectively. Assume that vector $\overrightarrow{t'}$ denotes the aspect-level polarity distribution of a summary $\mathcal{S}$, then its polarity distribution preservation ratio to $\mathcal{R}$ is defined as

$$\eta = \texttt{corr}(\overrightarrow{t'}, \overrightarrow{t})$$

where $\texttt{corr}(\cdot)$ denotes the Pearson correlation coefficient function. A value of $\eta \in [-1, 1]$ that is close to one means that the summary has well preserved the aspect-level polarity distribution of $\mathcal{R}$.

5.3 Baselines

We compare our methods, denoted by $\mathcal{S}_{\texttt{CCL}}$ and $\mathcal{S}_{\texttt{SCL}}$, with other three baselines. In order to avoid length-based bias,[4] we add constraints on the number of sentences selected so that the sizes of summary returned by each baseline are roughly equal to that of $\mathcal{S}_{\texttt{SCL}}$. For $\mathcal{S}_{\texttt{CCL}}$, we report the result when $\mathcal{B} = |\mathcal{S}_{\texttt{SCL}}|$ (denoted as $\mathcal{S}^b_{\texttt{CCL}}$) and the optimal result ($\mathcal{S}^*_{\texttt{CCL}}$) in terms of both aspect coverage and polarity preservation achieved by varying $\mathcal{B}$.

- Aspect-based sentence selection ($\mathcal{S}_{\texttt{a}}$): In aspect-based sentence selection, we assume that a set of aspects are given as inputs. Therefore, we read the manually labeled aspect lists as an input, group sentences towards the same aspect

[4] A longer summary is more likely to provide better information but is less concise.

into a same cluster, and select a number of representative sentences (i.e., a set of sentences that are most similar to other sentences in the same cluster) from each cluster C with probability $p_1 = \frac{|C|}{|\mathcal{R}|}$, which implies that for hot aspects, more sentences would be selected. The extraction is terminated when the size of selected sentences reaches the size of sentences selected by $|\mathcal{S}_{\mathtt{SCL}}|$.

- Position-based sentence selection ($\mathcal{S}_{\mathtt{p}}$): In position-based sentence selection, sentences are selected from the beginning and ending positions of each review document/paragraph, assuming that the locations are related to the likelihood of the sentences of being chosen for summarization [2].
- Ranking-based sentence selection ($\mathcal{S}_{\mathtt{r}}$): After computing the sentence graph, ranking-based sentence selection uses graph-based ranking techniques [30] to sort sentences in a reversed order based on their scores, and the top ranked sentences are selected. The number of selected sentences is equal to that in $\mathcal{S}_{\mathtt{SCL}}$.

5.4 Quantitative Evaluation

Firstly, we report the number of sentences in summary returned by $\mathcal{S}_{\mathtt{SCL}}$ and $\mathcal{S}_{\mathtt{CCL}^*}$ in Table 2. Note that we do not report the size of other summaries since they are either equal to or very similar to the size of $\mathcal{S}_{\mathtt{SCL}}$. In terms of concise, $\mathcal{S}_{\mathtt{SCL}}$ summary which is able to achieve 92 % compression ratio in the worst case, is significantly better than the $\mathcal{S}_{\mathtt{CCL}^*}$ summary.

Next, we study how the proposed method performs with respect to the aspect coverage ζ. The results are reported in Table 3. The baseline $\mathcal{S}_{\mathtt{a}}$ is supposed to maximize the aspect coverage and achieve 100 % coverage. However, with the usage of the probing probability p_1, some unpopular aspect is missing in $\mathcal{S}_{\mathtt{a}}$. Therefore, $\mathcal{S}_{\mathtt{a}}$ achieves only 92 % coverage on average. Regarding leader-based summaries, $\mathcal{S}_{\mathtt{SCL}}$ performs better than $\mathcal{S}^b_{\mathtt{CCL}}$, but slightly worse than $\mathcal{S}^*_{\mathtt{CCL}}$. This is understandable since the size of summary outputted by $\mathcal{S}^*_{\mathtt{CCL}}$ is much longer than that of $\mathcal{S}_{\mathtt{SCL}}$. Furthermore, from the results, we can find that aspect coverage of leader-based summaries ($\mathcal{S}_{\mathtt{SCL}}$, $\mathcal{S}^*_{\mathtt{CCL}}$, $\mathcal{S}^b_{\mathtt{CCL}}$) is comparable to that of $\mathcal{S}_{\mathtt{a}}$ on average. On some product domains such as Dell laptop and ipad, leader-based summaries are even better. Ranking-based method $\mathcal{S}_{\mathtt{r}}$, performs worse than both $\mathcal{S}_{\mathtt{a}}$ and leader-based summaries, but has much

Table 2 The size of summary

	Case	Laptop	iMac	ipad	Protector	Kindle
$\lvert\mathcal{S}_{\mathtt{SCL}}\rvert$	96	27	44	94	69	234
$\frac{\lvert\mathcal{S}_{\mathtt{SCL}}\rvert}{\lvert\mathcal{R}\rvert}$ (%)	3.35	5.76	7.76	2.63	7.24	1.07
$\lvert\mathcal{S}_{\mathtt{CCL}^*}\rvert$	183	104	40	303	108	544
$\frac{\lvert\mathcal{S}_{\mathtt{CCL}^*}\rvert}{\lvert\mathcal{R}}$ (%)	6.39	22.17	7.05	8.48	11.33	2.48

Table 3 Aspect coverage ζ comparison

Product	$\mathcal{S}_{\text{SCL}}$ (%)	$\mathcal{S}^{*}_{\text{CCL}}$ (%)	$\mathcal{S}^{b}_{\text{CCL}}$ (%)	$\mathcal{S}_{\text{a}}$ (%)	$\mathcal{S}_{\text{p}}$ (%)	$\mathcal{S}_{\text{r}}$ (%)
Case	94	95	88	100	88	76
Laptop	90	93	86	85	65	86
iMac	88	83	83	94	47	47
ipad	97	95	88	94	85	69
Protector	84	89	82	84	37	82
Kindle	94	98	91	97	88	91
Average	91	92	86	92	68	75

Table 4 Polarity preservation η comparison

Product	$\mathcal{S}_{\text{SCL}}$	$\mathcal{S}^{*}_{\text{CCL}}$	$\mathcal{S}^{b}_{\text{CCL}}$	$\mathcal{S}_{\text{a}}$	$\mathcal{S}_{\text{p}}$	$\mathcal{S}_{\text{r}}$
Case	0.93	0.99	0.91	0.78	0.92	0.84
Laptop	0.98	0.93	0.73	0.61	0.64	0.63
iMac	0.79	0.97	0.46	0.14	0.24	0.51
ipad	0.97	0.90	0.84	0.57	0.9	0.66
Protector	0.87	0.84	0.33	0.48	0.45	0.73
Kindle	0.85	0.97	0.76	0.68	0.76	0.80
Average	0.9	0.93	0.67	0.54	0.65	0.70

better aspect coverage than $\mathcal{S}_{\text{p}}$. These results indicate that the proposed methods $\mathcal{S}_{\text{SCL}}$, $\mathcal{S}^{b}_{\text{CCL}}$, and $\mathcal{S}^{*}_{\text{CCL}}$ perform well in terms of aspect coverage ζ.

Finally, we compare the performance of different methods for opinion summarization in terms of polarity distribution preservation ratio η. The goal of this experiment is to evaluate whether the summary generated by different methods can preserve the polarity distribution of each aspect of the original reviews $\mathcal{R}$. The results are shown in Table 4. As can be seen from the table, our proposed method $\mathcal{S}_{\text{SCL}}$, $\mathcal{S}^{b}_{\text{CCL}}$, and $\mathcal{S}^{*}_{\text{CCL}}$ obtain much better results than other baselines and can preserve polarity distribution of the original reviews in the aspect level. The Aspect-based sentence selection method $\mathcal{S}_{\text{a}}$ may select a number of very popular sentences but express redundant viewpoints towards a specific aspect, which results in that the polarity distribution of the selected sentences within an aspect may easily got skewed. Surprisingly, from the table we find that the Position-based method $\mathcal{S}_{\text{p}}$ does not perform worst in terms of polarity distribution preservation. A possible reason is that usually the first or last sentences in a paragraph/review are likely to express a viewpoint towards an entity, such as "Overall, 5 stars for the price!". As a result, the sentences selected by $\mathcal{S}_{\text{p}}$ can obtain reasonable performance in terms of polarity distribution preservation.

In the above study, both Tables 3 and 4 show that $\mathcal{S}^{*}_{\text{CCL}}$ outperforms $\mathcal{S}_{\text{SCL}}$ in terms of aspect coverage and polarity preservation. Since the CCL algorithm well preserves both aspect coverage and polarity distribution, we recommend the proposed opinion

summarization system to use the CCL algorithm when aspect-based summarization is required to be post-applied to sentence-based summaries before displaying to users.

5.5 User Study

In the previous section we investigated how different methods perform in terms of aspect coverage and polarity distribution preservation. In this section, we perform a user study to understand how useful the sentence selected by different methods are to actual users. An ideal way to conduct the user study is to let users select sentences for summarization manually as references and then evaluate the similarity between the references and different system summaries using ROUGE-N [24].[5] However, for our dataset, it is difficult to generate summaries manually especially for some product domain where the size of reviews is up to 21,948 sentences. Instead, we asked a number of humans to express their preference for one summary over another one. Each person is required to conduct 20 groups of rating and in each group two summaries of the same product are placed side-by-side in a random order. We did not ask users to rate $\mathcal{S}^b_{\text{CCL}}$ since $\mathcal{S}^*_{\text{CCL}}$ is consistently better than $\mathcal{S}^b_{\text{CCL}}$, as defined.

The results of judgment agreement and preference evaluation are reported in Table 5, where "agreement" is the percentage of items for which all raters agreed on a positive/negative/no-preference rating while "prefer A/B" is the percentage of agreement items in which the raters prefer either A or B respectively. As can be observed that the proposed methods are much better than the other baselines. More than 66.6 % comparison judges show that the Leader-based summaries ($\mathcal{S}^*_{\text{CCL}}$ and $\mathcal{S}_{\text{SCL}}$) are better than all the other baselines while the agreement is also up to 80 %. In

Table 5 Results of user evaluation experiments

Comparison (A V.S. B)	Agreement (%)	Prefer A (%)	Prefer B (%)	Equal (%)
$\mathcal{S}_{\text{SCL}}$ V.S. $\mathcal{S}_{\text{a}}$	86.0	72.4	16.3	11.3
$\mathcal{S}_{\text{SCL}}$ V.S. $\mathcal{S}^*_{\text{CCL}}$	75.0	70.1	12.0	17.9
$\mathcal{S}_{\text{SCL}}$ V.S. $\mathcal{S}_{\text{p}}$	84.0	73.2	13.4	13.4
$\mathcal{S}_{\text{SCL}}$ V.S. $\mathcal{S}_{\text{r}}$	85.3	89.7	2.0	8.3
$\mathcal{S}^*_{\text{CCL}}$ V.S. $\mathcal{S}_{\text{a}}$	80.0	67.3	15.2	17.5
$\mathcal{S}^*_{\text{CCL}}$ V.S. $\mathcal{S}_{\text{p}}$	86.4	66.6	12.2	21.2
$\mathcal{S}^*_{\text{CCL}}$ V.S. $\mathcal{S}_{\text{r}}$	84.4	88.7	2.0	9.3
$\mathcal{S}_{\text{a}}$ V.S. $\mathcal{S}_{\text{p}}$	65.0	56.5	34.5	9.0
$\mathcal{S}_{\text{a}}$ V.S. $\mathcal{S}_{\text{r}}$	62.0	64.7	28.7	6.6
$\mathcal{S}_{\text{p}}$ V.S. $\mathcal{S}_{\text{r}}$	68.1	45.3	48.4	6.3

[5] ROUGE-N is a popular toolkit which measures the quality of a summary by comparing it to other reference summaries using n-gram co-occurrence.

addition, users prefer the summary outputted by $\mathcal{S}_{\mathtt{SCL}}$ than $\mathcal{S}^*_{\mathtt{CCL}}$. One possible reason is that usually summaries generated by $\mathcal{S}^*_{\mathtt{CCL}}$ are much longer and hence result in lower scores in readability and conciseness. Therefore, we recommend the proposed system to use the SCL algorithm due to its good trade-off between conciseness and aspect coverage when the sentence-based summary is directly displayed to users.

For the remaining baselines, there is no obvious winner among them except that the Aspect-based approach $\mathcal{S}_{\mathtt{a}}$ is more preferred than the Ranking-based approach $\mathcal{S}_{\mathtt{r}}$. The reason may be that each baseline is designed to optimize a specific measure (e.g., ranking-based method is proposed to optimize the informativeness) while the user quality study is evaluated over a combination of criteria. In contrast, our proposed methods aim to select informative sentences by optimizing aspect coverage and preserving polarity distribution simulatively, which may be more desirable for users' demand.

6 Conclusions and Future Works

In this paper, we have developed an effective framework for informative sentence selection for opinion summarization. The informativeness of sentences is evaluated in terms of aspect coverage and viewpoints coverage. To this end, we have formulated the informative sentence selection problem as a community leader detection problem in sentence graph, where edges encode the term similarity and viewpoint similarity of sentences. Next, we have presented two effective algorithms to find the leaders (informative sentences) and communities (sentences with similar aspects and viewpoints). A set of systematic evaluation as well as quality evaluation verified that the proposed methods are able to achieve good performance.

Though the primary focus of this paper is opinion summarization, our approach is also applicable to other opinion mining problems. Therefore, one avenue for the future work is to exploit our sentence extraction method for other tasks such as spam review detection. In addition, in this paper, we conduct a set of empirical studies on product review data. In the future, we also plan to extend our methods to different domains such as twitter data, conversation, and political forum data.

Acknowledgments This work is partially supported by DARPA under grant Number W911NF-12-1-0034.

References

1. Ageev AA, Sviridenko M (1999) Approximation algorithms for maximum coverage and max cut with given sizes of parts. In: Proceedings of the 7th international conference on integer programming and combinatorial optimization, Springer, London, pp 17–30
2. Beineke P, Hastie T, Manning C, Vaithyanathan S (2004) Exploring sentiment summarization. In: AAAI spring symposium on exploring attitude and affect in text: theories and applications
3. Blair-goldensohn S, Neylon T, Hannan K, Reis GA, Mcdonald R, Reynar J (2008) Building a sentiment summarizer for local service reviews. In: NLP in the information explosion era

4. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
5. Bookstein A (1990) Informetric distributions, part i: unified overview. J Am Soc Inf Sci 41(5):368–375
6. Cheng J, Ke Y, Fu AWC, Yu JX, Zhu L (2010) Finding maximal cliques in massive networks by h*-graph. In: Proceedings of the SIGMOD. ACM, New York, pp 447–458
7. Danescu-Niculescu-Mizil C, Kossinets G, Kleinberg JM, Lee L (2009) How opinions are received by online communities: a case study on amazon.com helpfulness votes. In: Proceedings of the 18th WWW, ACM, New York, pp 141–150
8. Erkan G, Radev DR (2004) Lexpagerank: prestige in multi-document text summarization. In: Proceedings of EMNLP, Barcelona, Spain
9. Filippova K (2010) Multi-sentence compression: finding shortest paths in word graphs. In: COLING, pp 322–330
10. Freeman LC (1979) Centrality in social networks: conceptual clarification. Soc Netw 1(3):215–239
11. Ganesan K, Zhai C, Han J (2010) Opinosis: a graph based approach to abstractive summarization of highly redundant opinions. In: Proceedings of the 23rd COLING
12. Heerschop B, Goossen F, Hogenboom A, Frasincar F, Kaymak U, de Jong F (2011) Polarity analysis of texts using discourse structure. In: Proceedings of the 20th CIKM. ACM, New York, pp 1061–1070
13. Hirsch JE (2005) An index to quantify an individual's scientific research output. Proc Natl Acad Sci U S A 102(46):16569–16572
14. Hofmann T (1999) Probabilistic latent semantic analysis. In: Proceedings of uncertainty in artificial intelligence, pp 289–296
15. Hu B, Song Z, Ester M (2012) User features and social networks for topic modeling in online social media. In: ASONAM, pp 202–209
16. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings of the 10th ACM SIGKDD. ACM, New York, pp 168–177
17. Jin F, Huang M, Zhu X (2010) A comparative study on ranking and selection strategies for multi-document summarization. In: COLING (Posters), pp 525–533
18. Khuller S, Moss A, Naor JS (1999) The budgeted maximum coverage problem. Inf Process Lett 70:39–45
19. Kim HD, Ganesan K, Sondhi P, Zhai C (2011) Comprehensive review of opinion summarization
20. Kim SM, Pantel P, Chklovski T, Pennacchiotti M (2006) Automatically assessing review helpfulness. In: Proceedings of EMNLP. Association for Computational Linguistics, Stroudsburg, pp 423–430
21. Lerman K, Blair-Goldensohn S, McDonald R (2009) Sentiment summarization: evaluating and learning user preferences. In: Proceedings of the 12th EACL. ACL, Stroudsburg, pp 514–522
22. Li F, Huang M, Yang Y, Zhu X (2011) Learning to identify review spam. In: IJCAI, pp 2488–2493
23. Lim EP, Nguyen VA, Jindal N, Liu B, Lauw HW (2010) Detecting product review spammers using rating behaviors. In: Proceedings of the 19th CIKM. ACM, New York, pp 939–948
24. Lin CY, Hovy E (2003) Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Proceedings of the NAACL. ACL, Stroudsburg, pp 71–78
25. Lin H, Bilmes J (2011) A class of submodular functions for document summarization. In: Proceedings of the 49th HLT/ACL. ACL, Stroudsburg, pp 510–520
26. Liu J, Cao Y, Lin CY, Huang Y, Zhou M (2007) Low-Quality product review detection in opinion summarization. In: Proceedings of the joint conference on EMNLP-CoNLL, pp 334–342
27. Lu Q, Getoor L (2003) Link-based classification. In: Proceedings of the 20th ICML. AAAI Press, Chicago, pp 496–503
28. Lu Y, Zhai C, Sundaresan N (2009) Rated aspect summarization of short comments. In: Proceedings of the 18th WWW. ACM, New York, pp 131–140
29. Mei Q, Ling X, Wondra M, Su H, Zhai C (2007) Topic sentiment mixture: modeling facets and opinions in weblogs. In: Proceedings of the 16th WWW. ACM, New York, pp 171–180
30. Mihalcea R, Tarau P (2004) Textrank: bringing order into text. In: EMNLP, pp 404–411

31. Muthukrishnan P, Gerrish J, Radev DR (2008) Detecting multiple facets of an event using graph-based unsupervised methods. In: COLING, pp 609–616
32. Newman MEJ (2007) The mathematics of networks. The new palgrave encyclopedia of economics pp 1–12
33. Pang B, Lee L (2004) A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd ACL. Association for Computational Linguistics, Stroudsburg
34. Popescu AM, Etzioni O (2005) Extracting product features and opinions from reviews. In: Proceedings of the HLT and EMNLP. Association for Computational Linguistics, Stroudsburg, pp 339–346
35. Sabidussi G (1966) The centrality index of a graph. Psychometrika 31(4):581–603
36. Smith LM, Zhu L, Lerman K, Kozareva Z (2013) The role of social media in the discussion of controversial topics. In: SocialCom, pp 236–243
37. Taboada M, Anthony C, Voll K (2006) Methods for creating semantic orientation dictionaries. In: Proceedings of 5th ICLRE, Genoa, Italy pp 427–432
38. Taskar B, Wong M, Abbeel P, Koller D (2004) Link prediction in relational data. In: NIPS. MIT Press, Cambridge
39. Titov I, McDonald RT (2008) A joint model of text and aspect ratings for sentiment summarization. In: ACL, pp 308–316
40. Tsaparas P, Ntoulas A, Terzi E (2011) Selecting a comprehensive set of reviews. In: Proceedings of the 17th ACM SIGKDD. ACM, New York, pp 168–176
41. Wan X, Yang J (2008) Multi-document summarization using cluster-based link analysis. In: Proceedings of the 31st ACM SIGIR. ACM, New York, pp 299–306
42. Wang D, Li T (2010) Document update summarization using incremental hierarchical clustering. In: Proceedings of the 19th CIKM. ACM, New York, pp 279–288
43. Yu J, Zha ZJ, Wang M, Chua TS (2011) Aspect ranking: Identifying important product aspects from online consumer reviews. In: ACL, The Association for Computer Linguistics, pp 1496–1505
44. Zhu L, Galstyan A, Cheng J, Lerman K (2014) Tripartite graph clustering for dynamic sentiment analysis on social media. In: SIGMOD Conference, pp 1531–1542
45. Zhu L, Galstyan A, Cheng J, Lerman K (2014) Tripartite graph clustering for dynamic sentiment analysis on social media. CoRR abs/1402.6010
46. Zhu L, Gao S, Pan SJ, Li H, Deng D, Shahabi C (2013) Graph-based informative-sentence selection for opinion summarization. In: ASONAM, pp 408–412
47. Zhuang L, Jing F, Zhu XY (2006) Movie review mining and summarization. In: Proceedings of the 15th CIKM. ACM, New York, pp 43–50

Social Media Question Asking: A Developing Country Perspective

Hasan Shahid Ferdous, Mashrura Tasnim, Saif Ahmed and Md. Tanvir Alam Anik

Abstract The last decade has seen the emergence of the social networking sites (SNS) and researchers are investigating the useful applications of this technology in various areas apart from its recreational value. Ubiquitous presence of SNS has enabled us to obtain customized information seamlessly from our acquaintance. There have been many works that analyzed the types and topics of questions people ask in these networks and why. Topics like what motivates people to answer such queries, how to integrate the traditional search engines, and SNS together are also well investigated. In this research, we focus on the use of this technology in underdeveloped parts of the world and the new doors it has opened for its inhabitants in terms of obtaining information. Analyzing 880 status messages collected from a widely used SNS, we have observed that, unavailability and inadequacy of information on web in developing countries play a significant role to motivate users using SNS for information retrieval. Based on a structured survey on 328 persons, we have tried to emphasize the differences between social search and traditional web search. Our statistical analysis finds the correlations among different relevant parameters and provides insight that one might require to consider while developing any application for SNS-based searching.

Keywords Developing countries · Social query · Information search · Social networking sites

H.S. Ferdous (✉)
University of Melbourne, Parkville, VIC, Australia
e-mail: hasan.ferdous@unimelb.edu.au

M. Tasnim · S. Ahmed · Md.T.A. Anik
BUET, Dhaka, Bangladesh
e-mail: mashrura_cse@yahoo.com

S. Ahmed
e-mail: saif.ahmed@csebuet.org

Md.T.A. Anik
e-mail: tanviranik@gmail.com

Ö. Ulusoy et al. (eds.), *Recommendation and Search in Social Networks*,
Lecture Notes in Social Networks, DOI 10.1007/978-3-319-14379-8_10

1 Introduction

The supremacy of human race comes from their ability to think and ask questions. We encounter a wide range of information needs in our everyday life. These include questions or queries, recommendations about career development, factual knowledge regarding sophisticated technologies, rhetorical thoughts of life events, opinions about a major purchase, etc. For a long part of our history, helping one another in this quest was the only available way. Then we learnt to preserve, convey, and spread our knowledge through written and printed medium. The digital revolution over the past three decades has provided us with new power to store and maintain large collection of data in a tiny amount of space. Especially, the inception of search engines (SE) has enabled us to look into tremendous amount of information within seconds, a feat that our ancestors could hardly imagine about. These achievements lead many of us to believe that we are at the pinnacle of information search and retrieval, but as it comes out, it is hardly the truth. Many researchers are now wondering if history is repeating itself to bring us back to human intervention in information retrieval due to the emergence of social networking sites (SNS).

The past decade has seen the emergence of social networking sites (SNS), namely *Facebook*, *Twitter*, *Google+*, among many others. In this era of SNS, we are more connected with people around the world than ever before. Nowadays, it is no longer a source of entertainment and social connectivity only, it has paved a new way for information searching [1]. Apart from using the search engines that can merely use the already available information in the public sites crawled in its memory and some algorithm to search and index the results without much personalization, we can simply ask the members of our social network and get personalized and useful information that the researchers found quick, useful, and in many cases, more robust.

Using social networking sites as an information source have drawn the attention of the researchers for a while now. There have been many works that analyzed the types and topics of questions people ask in these networks and why. Topics like what motivate people to answer such queries, how to integrate the traditional search engines, and SNS together are also well investigated. In this research, we focus on a relevant but novel issue—how SNS search varies in developed and developing regions of the world and why. Analyzing 880 status messages collected from a widely used SNS, we have observed that, unavailability and inadequacy of information on web in developing countries play a significant role to motivate users using SNS for information retrieval. With established statistics of Internet usage, e-Governance, survey data, and our experimental data analysis, we have tried to emphasize the differences between social search and traditional web search in the developing country context and provided insight that one might require to consider while developing any application for SNS-based searching.

SNS provide users with source of information that is complementary to that provided by search engines. Search engines provide information that comes from ubiquitous source, i.e., web, in contrast to SNS that can provide objective data from a variety of sources on a variety of topics and is highly tailored to an individual. SNS

are connecting individuals to one another with whom they have a previously established offline connections or different degrees of relational closeness in online or in real life. Thus it is naturally likely that people turn to SNS as an efficient way to tap these connections for information seeking purposes [2]. Information obtained from SNS is also found to be highly trusted, as we know the individual behind the information too.

In this research, we will use Facebook as an example SNS, without losing any generality. With one billion monthly active users and more than half a billion daily active users [3], currently (as of December 31, 2013) it is the number two site in the world considering Internet traffic, according to Alexa ranking [4]. On an average, the users spend 10.5 billion min per day on it, make 421 million status message posts, 3.2 billion likes and comments, and have 140.3 friends in their Facebook network [5]. The average age of SNS users has also increased in recent years: among American Internet users, 70 % of 30–49 year olds, 51 % of 50–64 year olds, and 33 % of those 65 or older now have a profile on an SNS.

In this work, we emphasize this ubiquitous presence of SNS with special focus on developing regions of the world and see how SNS search has made significant changes in the way people access information here. We discuss the seminal works in this area in the next section. The problem of 'digital divide' is explained then along with the concept of less biased SNS world. Our experimental data along with methodology, interviews, analysis, and findings are explained in Sect. 5. Finally, we conclude after discussing and analyzing the implications of our survey data.

2 Related Works

Lampe et al. [1] analyzed how the use of Facebook has changed over time, using three consecutive years of survey data and thorough interviews with a few of the survey people. They reported that though the uses of the site remain relatively constant over time, but the perceived audience for user profiles and attitudes about the site showed differences over the study period. They find that patterns of use, perception, and attitude somewhat changed over the time. Their study, consistent with others, found that the number of friends and time spent on Facebook increased at first and then leveled off, which from interviews, suggested that new users spend time adding people as friends and getting used to the site. After a while, this behavior lessens as time is spent more seeing what is happening to friends instead of expanding their friend base. Also, new users are more likely to use Facebook to "find people to date" or "meet new people" than long-term users.

One of the important studies in SNS-based information search is done by Efron et al. [6], who identified that microblogging services like *www.twitter.com* are gradually becoming a popular venue for informal information interaction. They showed that question asking in microblogs is strongly tied to peoples' naturalistic interactions, which helped them to offer a taxonomy of questions in microblogs. They also showed that the act of asking questions in Twitter is not analogous to information

seeking in more traditional information retrieval environments, which contextualize these articulations through analysis of a large body of tweets.

Teevan et al. [7] discussed the types of information people used twitter to find, for example, breaking news, real-time content, popular trends, etc. This paper presented the systematic overview of search behavior on Twitter and differences with web search using questionnaire data along with analysis on query logs. They found that Twitter results included more social content and events, while web results contained more facts and navigation. Based on their study, they recommended that search engines could use trending Twitter queries to discover additional queries that have strong temporal components.

Lampe et al. investigated the Facebook user characteristics based on a survey of 614 people who used it to ask something [8]. They identified the perception of the relationships within network members as significant predictors of information seeking approach. They did not show any comparison between SNS and SE regarding obtaining any particular type of information. This question is addressed by Morris et al. [9], where they explored the pros and cons of using SNS as information source and compared user interaction when they search anything either on SNS or SE, involving 12 participants on their study. They find that 53 % of the users received quick responses from SNS and 83 % received responses eventually as well.

The effects of community size and contact rate were studied for synchronous social Q&A involving 402 Microsoft employees by White et al. [10]. The study analyzed the effects of these variables in terms of objective and subjective measures, and from the standpoint of askers, answers, and all members' general perceptions of utility. Every metric showed improvement with increased community size, including increased fraction of questions answered, asking effectiveness, answer quality, and answer ratings, along with a corresponding decrease in the time to receive an answer, number of users who were bothered by incoming questions, and fraction of the community that was interrupted.

Jeong et al. [11] compared the 'friend-sourced' answers obtained from SNS with traditional 'crowd-sourced' answers and concluded that 'friend-sourced' SNS systems are at least as good as their paid 'crowd-sourced' for providing answer to its users' queries. Liu et al. [12] analyzed the extrinsic factors that may influence the response rate in social question-answering process.

The type of questions and answers in SNS are investigated by Morris et al. [13] using a study of 624 people about their Facebook usage experience. They also explored the relationships between answer speed and quality, properties of participants (age, gender, and social network usage habits) and their questions (type, topic, and phrasing). Their study complies with the findings of many other researchers that while traditional SE is good for objective queries, SNS shows better results and interactions for subjective queries. There are many motivations for asking questions in SNS—among them the most important reason was the belief that people in our social network knows our *context* better, therefore may provide more relevant answers. Often people turn to SNS regarding objective questions if knowing the answer is not urgent, in the hope that some other friend in his network already knows the answer and will share his knowledge with him in due time.

Panovich et al. [14] evaluated the role of *tie strength* in question–response behavior as an indication of how close the relationship is—close friends are strong ties, while acquaintances are weak ties. In their study, they asked 19 participants to ask some technological recommendation questions through status messages. After the participants rated the received answers' quality, they compared that with a tie strength metric, and found that stronger tie provides better answers than weaker ties, in general. Also, they find that friends who have expertise in the question topic provide more trustworthy answer irrespective of strong or weak ties.

Farnham et al. [15] studied the suitability of *So.cl*: a web application that combines web browsing, search and social networking, designed for the purposes of sharing and learning around topics of interest by taking feedback from 32 college students. Their findings present the importance of social media for inspiring learning around new topics through social connections. They found the easy, lightweight integration of sharing around search in *So.cl* effectively fostered serendipitous and informal learning online.

Naaman et al. [16] examined 350 users' messages and some system data to understand the individual's activity using their own developed content-based categorization. Their analysis showed two common types of user behavior in terms of the content of the posted messages, and exposed differences between users in respect to these activities. But they did not address the relationship between social network structure and social influence to the type of content posted by users.

A controlled study conducted on 282 persons by Teevan et al. [17] analyzed effect of the factors: punctuation in status messages, scoping of audience, and precision on the response time, quality and quantity of response. Their key findings are that a higher portion of questions with a question mark received responses (88.1 % vs. 76.3 %, $p < 0.01$) and two-sentence questions received fewer and slower responses. They also noted that explicitly scoped questions resulted in better response.

Hecht et al. tried to combine the benefits of SE and SNS searching in their system named *SearchBuddies* [18], a system that responds to Facebook status message questions with algorithmic search results. They proposed two agents—Investigator (search on SE), that connects people with information, and Social Butterfly (Search on SNS), that connects people with other people who may have the desired information. After deploying their 'Socially Embedded Search Engine' on 122 users for 3 months, they believed that it provides highly relevant information in a social context. Horowitz et al. [19] presented *Aardvark*, a social search engine by which users can ask through email, message, voice, etc. Then Aardvark forwards that question to find the answer from someone expert and within asker's network, depending on the intimacy between them.

None of these researches investigate the difference in question–answer behavior in different parts of the world. Yang et al. [20] addressed this issue and identified some key differences between SNS search in the Western and Eastern cultural hemisphere. Their survey included people from US and UK representing the Western culture and people from China and India representing the Eastern culture. They concluded that people in the Eastern culture are somewhat more likely to use SNS for getting objective information than their counterpart and use it more often for the purpose.

They explained this phenomenon using existing and established knowledge from sociology study that Western cultures are associated with an analytic and low-context cognitive pattern, along with individualism, while Asian cultures are associated with a holistic, high-context cognitive pattern, along with interdependence and collectivist social orientation. Our initial findings match with them, except they did not include another possible explanation of this behavior—the existing web infrastructure deficit in the developing and undeveloped countries, commonly known as the *Digital Divide*. In our work, we will elaborate on this explanation.

3 The Divided World

The term 'Digital Divide' indicates the difference in technological advancement between the developed and developing/undeveloped parts of the world. Computers and other computing devices are essential commodities for the people in the developed region for the past two/three decades and their web presence is ubiquitous nowadays. Recent explosion in the smart phone usage has enabled virtually everyone to remain connected to Internet round the clock. Nearly all the governmental and business services have their information published and updated in the web. Traditional search engines in that respect are very effective in capturing the required information as it is already there in the Internet.

The scenario is quite opposite in the other parts of the world where the web culture has not flourished yet. If we focus on the Southeast Asia region as an example of the developing part of the world, we can see from UN survey 2010 [21] that the average e-Governance ranking of the 8 countries in this region is 134, way beyond the developed regions. According to Ref. [22], about 8–10 % people in this region have access to Internet. Even that is after the growth of Internet users in recent years, and the overall web presence is not good yet. Many important governmental and nongovernmental institutions do not have their information in the web and often do not update their information regularly, if there is any.

The problem is twofold. First, people in this developing region cannot find the required information from the web using traditional search engines as it is beyond its capacity to show any result that is not already in the web. Second, as the Internet culture has not flourished yet, many people are not used to search information in the web, or do not know how to find the right information if there is a lot of different search results. Though the governments in these countries are trying to eradicate this digital divide, it is proved as not that easy. The world remains *divided* and probably will remain so for a long time from now.

4 The Unified SNS World

In this section we will investigate the interaction of people from these undeveloped countries in the Internet. We consider 'Bangladesh' as representative country from Southeast Asia to provide some data on this. Bangladesh is ranked 3rd among the 8 countries of this region in the e-government ranking. Despite the efforts of the government to provide e-services to its citizens, the web presence of different government and nongovernment institutions is quite low. Internet access is available to only 5 % of her citizens and many of those who have access to Internet use it seldom. But if we consider the SNS presence of the people in Bangladesh, they are not far behind [22, 23].

There has been dispute regarding the total number of Internet users in Bangladesh. But despite the dissimilarity about the total number of Internet users from different online sources, it is noticeable that the ratio of the total number of Facebook users to Internet users from all the sources are close and roughly 43 % of the Internet users in Bangladesh use Facebook. If we compare this ratio with other countries in the world (Table 1), we can see that the ratio is good enough. A significant part of her Internet users are SNS user too.

This connectivity among the users has paved a new way for information gathering and sharing for the people of developing countries like Bangladesh. SE cannot give them the data that is not there in the web, but through SNS, their query can reach hundreds of the people of their acquaintance, and as Yang et al. [20] has already mentioned, they are traditionally encouraged to share their query with others. Through SNS, we can obtain information that others already know, and clarify information that we can find in the vast amount of data in the web. This is a unique opportunity for the people in these regions, which was never there before. Though it is not the end of 'digital divide' mentioned earlier, but we are getting a bit closer to unify the world in terms of information searching and retrieval capacity.

Table 1 Internet and Facebook usage analysis (All figures are in millions or percentage)

Country	Population	Internet user	Ratio with population (%)	Facebook user	Ratio with internet user (%)
Australia	22.8	17.9	78.3	11.7	65.7
USA	314.8	243.8	77.4	168.6	69.2
UK	62.3	51.2	82.2	33.8	66.0
Nepal	26.6	2.7	10.3	1.9	69.2
India	1210.2	125.0	10.3	60.6	48.5
Pakistan	181.3	15.9	8.8	7.6	47.7
Sri Lanka	20.3	3.2	15.6	1.5	46.2
Bangladesh	152.5	7.5	4.9	3.2	42.6

5 Experimental Data

Our data collection process had three phases. In the first part, we made a request for volunteers through our research group, from which we selected 10 enthusiastic participants from two universities in Bangladesh. All our participants had more than 150 friends in their Facebook profile (average 270) and uses Facebook regularly in their day-to-day life. Our participants had many of their friends in common, as they belonged to different levels in two institutes. In total, we could observe 1362 unique Facebook profiles through these 10 volunteers.

The volunteers were instructed about the data collecting process. They monitored the data stream in their Facebook home pages *passively* for questions asked through status posts and recorded those status posts with responses after a couple of days of making that post. This was required as Facebook does not show the exact time stamp of the comments after a few hours of making the original post. We collected data for about 8 weeks and received 880 of such queries. Then we analyzed those questions according to the categories mentioned by Morris et al. [13]. We tried to search answers for those questions using traditional search engines and compared them with the answers obtained from Facebook. We are still gathering more data, so the explanation provided in this section are not claimed as complete. But it should give some indication, emphasize our logic, and provide future directions for work.

We analyzed each of our 880 data sheets and summarized it into a table, which was then imported to a relational database management system (DBMS). We applied different sql techniques to compute the statistical mean, variance, z-score, p-value, and chi-square tests. We will present these data along with their implications in the following sections. We compared our results with that obtained by Morris et al. [13]. The survey participants in Morris et al. [13] were all employees at Microsoft, 72.8 % full time and the remaining were university students working as summer interns. Male and female ratio was 3:1, 68.1 % of their respondents were aged between 18 and 35 years.

In our second phase, we choose 10 participants using our already collected data. Five of them have asked at least one question in the past one month, while the rest have responded to at least one query made in Facebook during that period. We tried to investigate the motivation behind using social network as an information source and the inspiration that worked behind answering in it. Our interview data strongly supported our previous findings and also supported the findings made by Morris et al. [13]. In the third phase, we conducted a structured survey with students of some private and public universities in Bangladesh and analyzed those data.

Table 2 shows some analyses of our obtained data. The data has good similarities with the data obtained by others. Specially, like Yang et al. [20], our data also indicates that people in the eastern culture asks less subjective queries than people in the western countries. However, unlike many other works, our study shown later finds that significant part of the queries is related to finding factual information. When we analyzed the queries of such kind, we could understand the reason. Though these questions are objective and have definite answers, the users could not find the

Table 2 Question types and response analysis

Question type	Average first response (min)	Average total response	Appropriate answer	Time required to search through SE
Recommendation	8.5	6.2	Choice with reason	About 30 min searching
Opinion	4.7	9.5	No defined answer	No defined answer
Factual knowledge	7	6.9	Accurate in the 91.3 % cases, the rest are unanswered	No information for 56.5 % queries, about 5 min for others
Rhetorical	5	12	Not applicable	Not applicable
Invitation	4.2	15.5	Min. 1 positive reply	Not applicable
Favor	5.1	7.1	Min. 1 positive reply	Not applicable
Social connection	5	15	Yes	SE were not suitable
Offer	4.3	8.2	Yes	Not applicable

information in the web, and thus turning to SNS was the only option, aside contacting specific persons for it. As indicated by Morris et al. [9], people often do a Google search before asking anything through SNS, probably this was the case with our queries too. But the ratio of such queries is quite high in this region and considering the fact that web culture here has not expanded that much, it was somewhat expected.

When we analyze and compare our question types with that in Morris et al. [13] in Table 3, we can understand the validity of our claim. The queries asking about factual information comprises of the highest portion in our collected data, also supported by its high z-value (1.561). We can also see that there are significant differences in the

Table 3 Comparison in question type analysis with Morris et al. [13]

Question type	Percentage in Morris et al. [13]	Percentage in our data	z-score
Recommendation	29	7	−0.663
Opinion	22	20	0.914
Factual knowledge	17	25	1.561
Rhetorical	14	12	−0.042
Invitation	9	3	−1.128
Favor	4	18	0.605
Social connection	3	13	0.036
Offer	1	2	−1.284

Table 4 Comparison in question topic analysis with Morris et al. [13]

Question topic	Percentage in Morris et al. [13]	Percentage in our data	z-score
Technology	29	24	1.675
Entertainment	17	24	1.624
Home and Family	12	13	0.363
Professional	11	11	0.105
Places	8	5	−0.641
Restaurants	6	0	−1.182
Current events	5	11	0.105
Shopping	5	2	−1.001
Ethics and Philosophy	2	6	−0.383
Miscellaneous	–	4	−0.666

ratio at which people seek information regarding recommendation, invitation, favor, and social connection—supporting the arguments presented by Yang et al. [20]. A chi-square test value of 32.8 for $\alpha = 0.05$, which is far above the expected value of 14.067 also supports that our hypothesis is correct.

The question topics shows more similarity when we compare it with Morris et al. [13] indicating that people here face similar sort of queries in their daily life like their western counterparts. From Table 4 we can see that in most categories, our data shows considerable similarity with the data gathered by Morris et al. [13]. This observation is also supported by the chi-square test value of 14.4, which is less than 15.507 (from chi-square table) for $\alpha = 0.05$.

Then we have analyzed our data to identify if the factual queries posted by the people around us can actually be answered by searching in traditional search engines. We found that answers to 79.57 % of such queries cannot be found through SE, whereas 69.79 % of them got satisfactory answer through Facebook. Our analysis shows that 90.43 % queries got at least one answer and 20.64 % gets answer which might not be adequate or are unsatisfactory.

There are many queries on different topics that we could not find specific answer in the web. Some examples were like "When is the next performance by Shironamhin/James?" (two popular bands in Bangladesh). In the developed countries, we can expect that the music providers keep record of their future events and update it frequently. But here in Bangladesh, we could not find any specific site maintained by them. But when people asked it in Facebook, they got the information almost instantly (within 5 min).

Fig. 1 Example of question–answer in Facebook. Here a nontrivial question is asked that provoked discussion and important suggestions for the asker. Also the question was set in the local context where people from that locality could use their daily experience in replying to it

Another of the interesting queries and responses was about the traffic situation in a particular day. A person was on a very tight schedule to attend a workshop in Dhaka, Bangladesh. He was supposed to land in Dhaka Airport at 8:00 am, and his speech was scheduled at 10:00 am in front of the government dignitaries. So he was asking people in this locality about possible real-life traffic scenario during that time, describing the challenge he has to face. This kind of traffic information for Bangladesh is not available through Google map or any other service. But his friends could make valuable comments (Fig. 1), including an effective suggestion to get a front row seat while taking boarding pass, so that the queue in front of him in the immigration remains small. During interview, he pointed that this is one of the reason he prefers to ask such questions in Facebook as it may show unorthodox but unexpectedly useful solutions.

Another interesting query we find was about "Does *xx* University publish any journal?" Using Internet, we were able to find three journals published from that university. But in Facebook, the comments contained information about six journals. We contacted the relevant departments to verify that the information from those Facebook comments were accurate indeed. Those journals being local hard copy only had no online presence, and thus quite hard to find using search engines (Fig. 2).

Fig. 2 Another example of question-answer in Facebook. This is one of the examples showing that not all person has the same skill for searching in the web. So the information, though available in the web and could be found using search engines, the question setter was not able to do that (mentioned in the 3rd comment)

Local information is another kind of information that people seemed to seek through SNS. Queries like "Has there been any accident in *xx* Road?", "Do we have class test tomorrow?", "What movies are now showing in *xx* cinema hall?", etc., are such examples. These queries are answered promptly by friends in the SNS, but we could not find answers to them through searching the web (Fig. 3).

We also analyzed the response time of queries asked through SNS. We analyzed the queries and responses to record the time required for first response and for obtaining sufficient or satisfactory response. Both of these two data showed similar pattern. More than 40 % of the queries got some response before 20 min and about 70 % got the first response before 100 min had elapsed (Fig. 4). It also shows that queries that had not got the response by 10 h have very low chance of getting it later. The analysis data for reasonable response time shows similar pattern too (Fig. 5) with majority of the queries getting expected response by 2 h.

Fig. 3 Another example of question–answer in Facebook. The question was asked in Bangla, written using English alphabets in phonetic form. It asked about the price and availability of micro SIM cutter in a local market

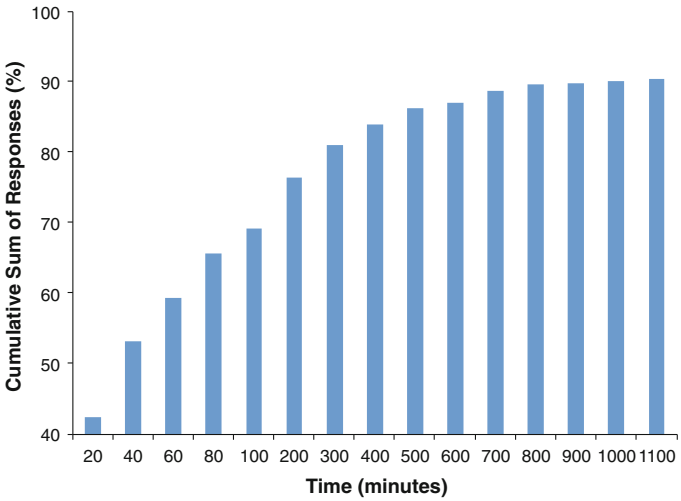

Fig. 4 Cumulative first response time versus time

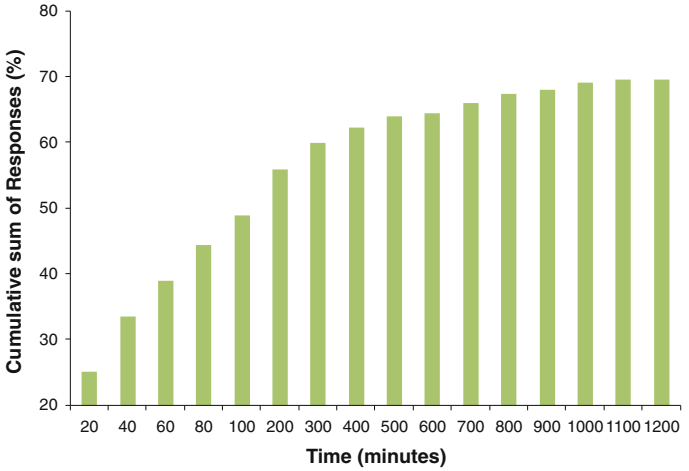

Fig. 5 Cumulative reasonable response time versus time

6 Interview Data

Ten participants were selected through personal connection with the researchers, each being a Facebook friend of at least one of us. Five of them have asked at least one question that we used in the first part of the study and five of them answered at least once. The interviews took place at a time and place convenient for the participants. We took shorthand notes during the interviews, which were then shown to the participants with no significant objection received. All the transcript of interviews and researcher notes was coded based on grounded theory approach, in order to find the main themes. In this section, we present a qualitative analysis of these interviews along with the main themes we found.

6.1 Local Information Is Limited

A lot of the queries analyzed in the previous section have some local origin, as emphasized by the amount of questions related to factual information and current events. When we asked our participants about why they have chosen Facebook to ask the questions, all of them agreed to the fact that traditional search engines do not satisfy many of their queries. While dealing with objective questions, they usually go through Google first. When they cannot find the information there, or are uncertain about the validity of it, they turn to Facebook to get the answer.

> When I could not find it in the web sites, I was not worried much. I have more than 400 friends in Facebook. Some of them must know it.

This firm believe on the social connection is seen in many cases. Even in the face of an emergency, when one of our participants needed to contact an army base station to know the whereabouts of her dear ones and found that the contact number provided in their web sites are not updated, she did not lose hurt. She went for help through her social network saying it was urgent and got the correct phone number within minutes. It provided her additional benefits too, as identified later in this section. Later she recalled,

> At some point, I was quite at a loss about what to do. Then I thought, many people among my Facebook friends are related to military services. Some of them should know the updated phone number, and it worked!

This faith on the availability of information from the social network has shown some issues too, discussed in the next section. In general, SNS has shown itself to work fine as an alternative source for local information, which is often either not available, or not updated regularly in Bangladesh.

6.2 Over Dependency on SNS

A person made a query about the location of the service center of a particular mobile operator. He got prompt reply from his friends. That information was available in the web and could be found easily. In fact, his friends have searched it for him and gave him the answer. When we asked him about it, he told us that this information was not urgent for him, and as he passes a lot of time using Facebook, he just made a post in the hope that someone may know it personally. He did not expect that this information is already available in the web and can be searched for. This shows that there is a gap in understanding the flourish of web technology in this region and often people are not aware what have changed around them in the past decade.

This has been a challenge for developing e-Services in developing countries like Bangladesh. The e-Services produced by the Government are used by too few people to make it a success. Identified by earlier research [24], though the Government in investing much resource and efforts in developing e-Services, many of its populace are not aware of these services and they do not believe that the computerized services are easier than previous systems. So in almost all cases, the Government is forced to keep both the legacy system and present e-Services, causing much more expense than earlier days.

It is relatively easy to get the answer from SNS, and we have seen a lot of people doing so despite the availability of information in the web. Their friends did provide the answer out of good gesture, but one of our participants got rebuked by his teacher.

> I got the answer, with a warning to Google it before I ask the community. But it was from my teacher, you know.

6.3 Motivation Behind Replying

Strong ties like close friends, work peers, neighbors are more encouraged to reply to queries in Facebook, supporting the earlier research works. Another important motivation for replying is to make a positive introduction of oneself to the asker. People are often more motivated to answer the queries made by their seniors, or someone with which they want a more positive relation with. And of course people often do it selflessly for a friend, or to show others about his expertise in the relevant topic. It is in the nature of human beings to help others, and that will remain as the driving force behind the success of SNS search.

There has been many research works that emphasized on the identification of expertise among the friend base to direct any social query. But we should remember that tie strength plays a major role in social networking sites as well as real social life. All 'friendships' are not equal, hence any algorithm to exploit this friend base in SNS will require to fine-tune the balance between social tie and expertise to effectively spread the query.

6.4 Social Connection Is Far More Significant Than Search Results

> People I know are so supportive and caring; not only I got the answer to my query, many of my friends later inquired about the wellbeing of my mother during the next weeks.

Answers to our queries are important of course. But the side effect of posting some queries often provide unique social resources and benefits, which we find is far reaching the expectations of the asker. For example, one of our participants asked for recommendation about some dentists in his locality. What happened next was well above his expectations. He obtained several name and addresses within the hour, of course, but there were a lot of people making queries about his well being too. When he confirmed that he is looking for a dentist for his mother, he obtained a lot of social support from his Facebook friends (many of which are his offline friends too).

Another of our participants jocularly made a point:

> Google provide me answers; Facebook provided me food, lodging, and company of a close friend during my visit to USA. Now, it is something hard to beat by a search engine.

He happened to go through Hong Kong in transit to his final destination in USA. He felt confused after reading the requirements for a transit visa. He sought the help of his friends who had similar experience. After some discussion in that thread, he came to know the real requirements and alternate scenarios. Then, from that post, some of his friends living in USA got to know that he is going to visit near their place and invited him to stay with him during the visit. Finally, he had a pleasant visit in addition to obtaining the information he required to know.

Similar things were observed with another participant asking for contact number in case of an emergency. Many friends came to know about her personal events from her social network and provided much needed emotional support.

We can consider another case of social empathy and support. When one of our participants burned an electrical equipment in his home and tried to figure out what electronic part he has burned in the circuit board, he uploaded a picture of it with his question. He asked, "The blue thing, is that an inductor?" The first reply came within seconds said, "that's the blue pill which keeps you in this matrix" (alluring to the famous movie 'The Matrix'). Despite his misfortunes, he could not resist the humor.

> I was badly looking for the answer, yet the first reply got me off-guard, and I could hardly stop laughing.

The requirement of considering social media question asking (SMQA) as a 'social' process, where the social connections play a much bigger role than being merely an information source is emphasized both in our experimental data and interviews.

6.5 Stray Discussions

Not all queries were fruitful. Some often provoked discussion in quite unwanted ways. When one of our participants asked for companion to some work, the discussion went astray. It ended in some political squabble, loosely related to that particular work. Though he did not took this matter seriously, saying "The discussion went to a different direction, but enjoyable anyways," because all who made comments are his friends, it shows some unexpected potential danger in exploiting social media for everyday queries.

6.6 Challenge in Contextual Interpretation

All the existing systems (for example, [18]) designed to provide automated answers in social media has largely failed due to contextual misinterpretation, providing irrelevant links as the answer. One big reason behind this discrepancy is the incomplete information provided in the query, which is often understood by the people from implicit contextual information or previous knowledge about the asker. One example we considered is this, where the asker posted: "Is there any place to get StarBucks coffee in Dhaka?" People came to understand immediately that he is not asking about Starbucks coffee outlet, because there is not any in Dhaka, the capital of Bangladesh. They interpreted the query that he is looking for Starbucks coffee beans rather than the brand outlet and answered accordingly. There were many such examples providing partial or very little information, specially about local or current events, thus making the interpretation of context very challenging.

7 Survey on SNS Q/A Usage

We have also conducted a structured survey on students from different private and public universities in Bangladesh to understand the use of SNS in getting information. These people constitute a significant portion of the populace that use social networks and other technological tools. The results show interesting opinions about SNS Q/A behavior. We have conducted both online and offline survey using the same questionnaire in English, which can be found in Ref. [25].

We have collected 328 responses in total, all of which are undergraduate level students. 93 % of our participants are from the age group of 18–24 years. The male–female ratio was not equal, 78 % being male. This is the common and expected ratio of male and female students in the undergraduate level in Bangladesh. 98 % of our survey participants were unmarried. We will see if these demographic properties have any impact on their Facebook usage pattern in later parts of this paper. However, these population does not represent all the Facebook users of Bangladesh, and the following analysis is not conclusive to be generalized, but will provide a good indication of the subject matter.

Almost all of our participants (99.1 %) have their own Facebook account and majority of them are using it for more than 2 years (Fig. 6). Majority of them have about 201–500 friends in Facebook (Fig. 7). When asked about how frequently they update their Facebook status (Fig. 8), most of them said they hardly update their status (64 %) or less than 3 times per week (22 %). 73.8 % of our participants have asked some question or opinion through Facebook, 23.5 % have never used Facebook for the purpose and 2.7 % of the participants did not reply to that question.

It appears from Fig. 9 that posting queries through Facebook is not a part of its day-to-day usage for the participants as majority of them (52 %) hardly post any question through Facebook and 25 % of them post less than three questions per week. However, the response time for queries posted in Facebook is quite good, as emphasized in Figs. 10 and 11. This phenomenon is elaborated in more details in later parts of our survey. We can see that more than 50 % of the participants expect to

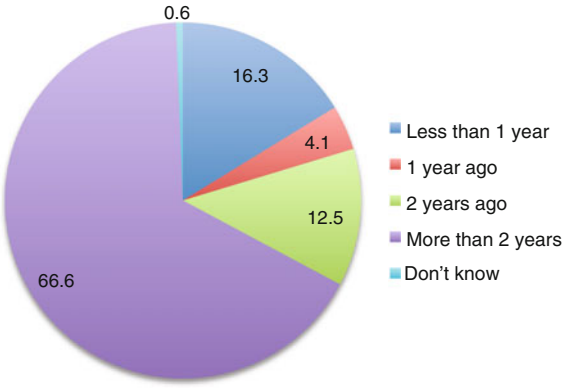

Fig. 6 Percentage of Facebook usage period

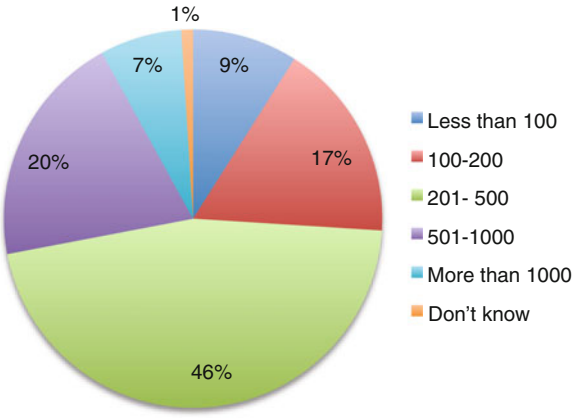

Fig. 7 Number of Facebook friends

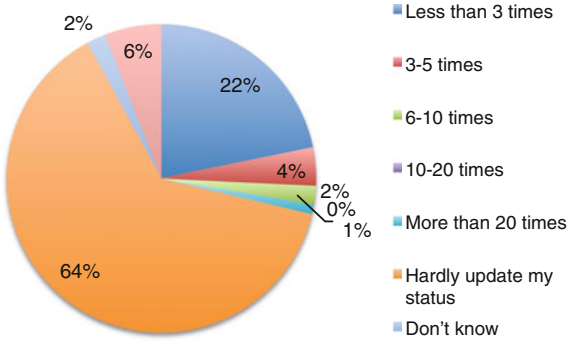

Fig. 8 Facebook status update (per week) statistics

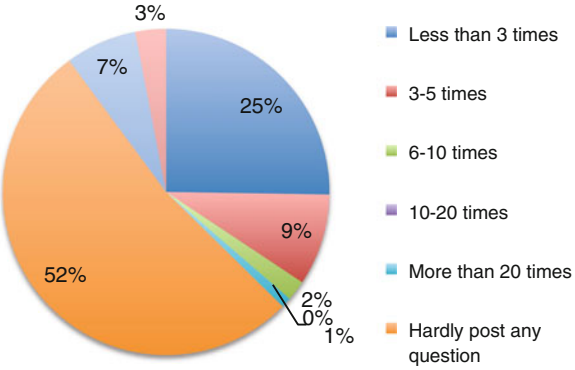

Fig. 9 Frequency of posting query in Facebook

get the first response for their query by 10 min only. When asked about the amount of time to obtain a satisfactory response, majority of the responses (28 %) were that 'I do not remember' and about 45 % of them get satisfied with the responses obtained within 30 min of making their query.

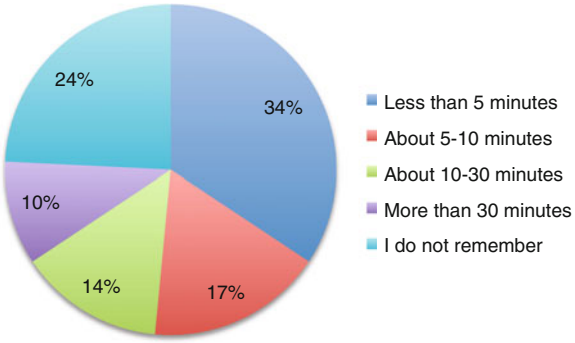

Fig. 10 First response time for the queries

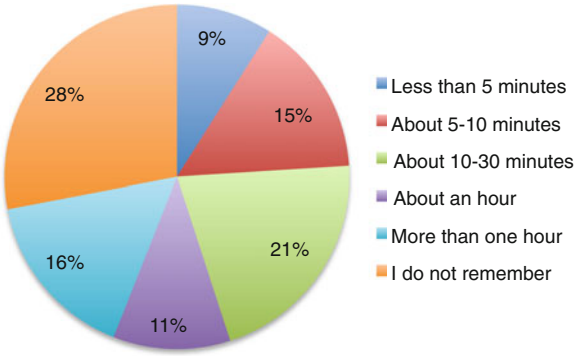

Fig. 11 Satisfactory response time for the queries

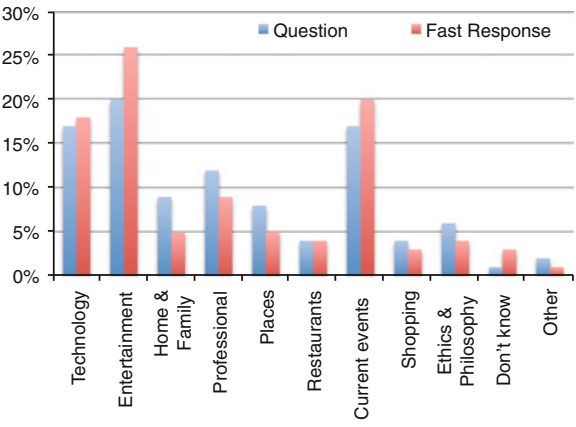

Fig. 12 Question topic and response preference

We also asked about the topics they post questions about. Three major areas are technology, entertainment, and current events (Fig. 12). They are also the topics where they get quick answers from their friends, as depicted in Fig. 12 too. About 67 % of the people provided opinion that they get satisfactory response to their queries from friends most of the times, if not always and 26 % people get it 'sometimes' (Fig. 14).

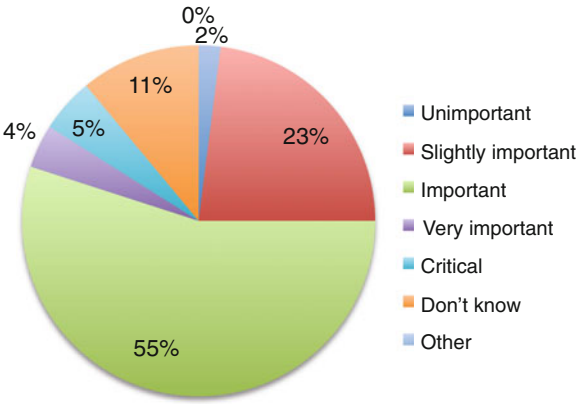

Fig. 13 Rating of the responses received

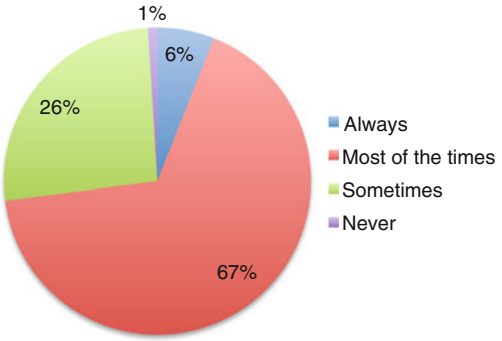

Fig. 14 Ratio of satisfactory response

Again, majority of the participants (55 %) rate the answer they get from their friends as *important* (Fig. 13). 52 % of the participants said that they also have obtained valuable information *many times* from the questions other people posted, whereas 35 % have said 'a few times' for the same.

When asked about the reason behind posting questions in Facebook, the most popular two reasons came out as 'it gives faster response' and 'it is not available in websites' (Fig. 15). Other popular options were that it is easier than searching and we can get more trustworthy answers from our friends.

Then we tried to compare the results people get from SNS and traditional search engines. Though 30 % of the participants agreed that both of them provide similar results, 47 % of them are not so sure about it and said that the results might differ and SNS can add some additional information or perspective to the problem at hand. Majority of the people (56 %) will choose Google for information searching, but 40 % said that they would decide either Facebook or Google depending on the query under consideration. In case they have limited time or bandwidth, 67 % will use Google, 27 % will decide depending on the query, and only 5 % will use Facebook solely. At the first sight, this data seemed contradictory to our original understandings that people use Facebook for informational purposes a lot more than that. However, our

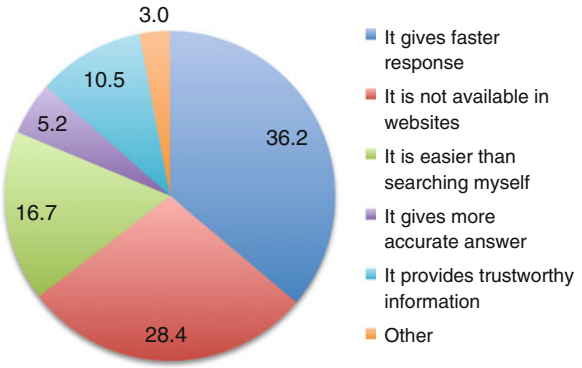

Fig. 15 Reason behind using Q/A in Facebook

survey population is the university students, most of them use computing devices for many tasks during their everyday life. So, they are constant users of the search engines, and considering the overwhelming number of queries they make during the whole day, these ratios are probably not misleading. If they do not get satisfactory information from Facebook, most of them (46 %) will ask someone personally about it, almost a similar percentage of people will search using traditional search engines (42 %) (Fig. 16).

To understand the advantage of SNS over SE, or vice versa, 27 % people think that Google enjoys the advantage that it can crawl a vast amount of data that is far beyond the capacity of human being. But Facebook queries are more tailored to my needs as our friends understands the context of the question (35 %) and through Facebook we can get the information that is available not in the web (34 %)—both factors strengthening Facebook as an information source.

Most of the people response to a query they see, at least 'sometimes' (Fig. 17). The main motivation is being helpful to others (Fig. 18). To assist others in their friend list, 49 % of the people often do a search themselves to provide an answer,

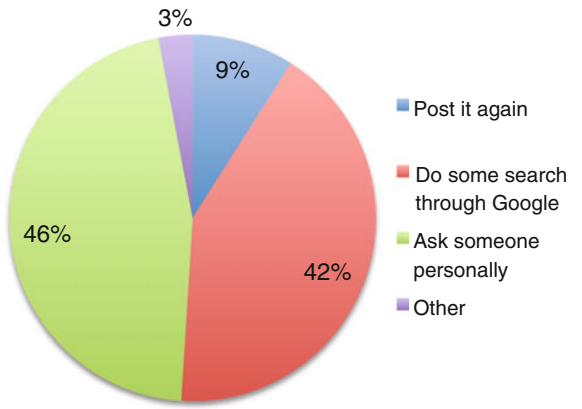

Fig. 16 Actions when no satisfactory answer is obtained from SNS

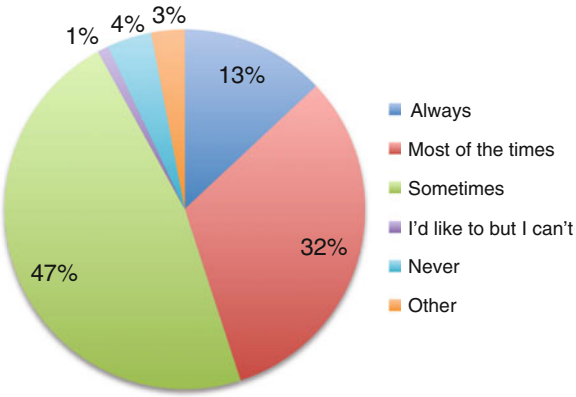

Fig. 17 Response to others' queries

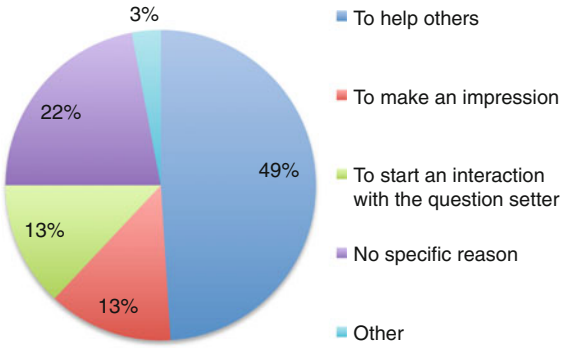

Fig. 18 Motivation behind responding to others' queries

while 24 % said that they ask another friend personally to know the answer and let the asker know about it.

8 Interrelation Among the Parameters

We used chi-square test of independence to find out interdependencies among different parameters based on demographics and SNS usage pattern (Table 5). Most of the tests showed independent behavior, for $p < 0.05$. When the test results showed dependency, it means, at least one of the samples deviate significantly from the other samples. The test does not identify where the differences occur or how many differences actually occur, so we calculated the z-scores of different samples for that query to see the dependencies or anomalies among them.

Our analysis shows that though the gender ratio among the participants is not equal (78/22), their Facebook usage patterns, specially the first/reasonable response time are independent of their gender. And though the lifetime of the users' Facebook account has no co-relation with response time and number of times they repost their

Table 5 Chi-Square test of independence on SNS question asking and answering behavior

Null hypothesis	Independent of–	Dependent?	Comment
Type of Question	Time to get first response	No	
	Time to get satisfactory response	No	
Gender	Time to get first response	No	
	Time to get satisfactory response	No	
	Type of question asked	No	
Length of Facebook use	Time to get first response	No	
	Time to get satisfactory response	No	
	Importance of Facebook as information source	No	
	Number of query update in search engine	Yes	People who uses Facebook longer are likely to modify their query and search again
	Preferred source for information search	Yes	People who have been using Facebook for 'More than 2 years ago' shows slightly more inclination to use Facebook as an information source than others
	Type of questions asked	No	
Number of friends	Time to get first response	Yes	First response time decreases slightly with the increase in number of friends
	Time to get satisfactory response	No	
	Importance of Facebook as information source	No	
	Number of query update in search	Yes	People who have more friends modify their query more than others
	Preferred source for information search	Yes	Though Google is the first choice as an information source for all groups, people with moderate number friends (201–500) chooses it more than others
Frequency of status update	Time to get first response	No	
	Time to get satisfactory response	No	
	Importance of Facebook as information source	No	

(continued)

Table 5 (continued)

Null hypothesis	Independent of–	Dependent?	Comment
Frequency of posting questions	Time to get first response	Yes	Frequent askers get response quickly
	Time to get satisfactory response	No	
	Importance of Facebook as information source	No	
	Type of questions asked	Yes	People who 'Hardly post any question' asks about 'current event' or 'entertainment' more than others
Response to others' questions	Importance of Facebook as information source	No	
	Preferred source for information search	Yes	People who always reply to others choose Facebook, but who 'sometimes' response to others choose Google more

queries, it is co-related with the information source they use for asking queries. People who have used Facebook for more than two years have considered using 'both Facebook and Google, depending on the query' more than any other sample populations.

The length of Facebook usage or the number of friends has no impact on the topics of their posted questions. However, the more friends they have, the quicker the first response to their queries comes. But interestingly, the reasonable response time is independent of the number of Facebook friends. Also, according to our analysis, those who have more friends, prefer to use 'both Facebook and Google' as an information source more than the other sample groups.

The analysis shows that there is no apparent relation between the frequency of normal status update with first/reasonable response time, preferred source of information search, or question topics. However, frequency of question posting has some dependency with the first response time and question topics—people who posts question frequently gets their first response quicker than others and people who seldom posts queries are more interested in 'current events' and 'entertainments.' We also found that the topic of question has no impact on first/reasonable response time or preferred source of information. Those who responses to others' queries 'most of the times' or 'sometimes' choose 'both Facebook and Google' as their preferred information source.

9 Design Implications

Combining the strength of SNS and SE is an ongoing research topic with yet any good usable solution to appear. Designing such a solution has many challenges and requires accurate understandings of the users' demands from these systems along with their responses to them. There has been very few works about cross-cultural studies on this topic, and our research emphasizes the importance of it before developing any successful platform to combine SE and SNS.

First of all, any such system needs to be aware of the cultural differences across the globe; it should not assume that one shoe fits all. Challenges of information validation in the underdeveloped regions will be of great importance due to lack of available information in the web. Depending on the friend base of a person also poses challenges, the issues of strong and weak ties need to be investigated and understood well. And the *Dunber Number* phenomenon comes into consideration, as our study shows that having higher number of friends does not ensure quicker response. So randomly selecting from the friend list may not work as expected. The complex interpersonal relationship and ties need to be understood for making it a success.

One novel finding of our research is that question topics in different cultures and regions of the world do not vary much, indicating that people all over the world have similar queries in their day-to-day life and search for answers. However, question types vary significantly and anyone designing for SNS search applications may require keeping that in mind. We also identified the challenges in contextual interpretation of questions in social media, requiring nontrivial mechanisms for developing automated response systems in social media question asking.

10 Conclusion

In this work, we have focused on differences of the SNS searching habits in different regions of the world based on their economic context. We showed that the motivation for SNS search in developing regions could be quite different than in the developed parts. The lack of information availability has played a major role in peoples' turning to SNS to get answer than from traditional search engines. Whether other factors like culture, religion, etc., play a significant role alongside these factors remain as a major research challenge. We are now working on developing a Facebook app using this information that will assist the users to obtain real-life information from Facebook in a better way than now.

Acknowledgments This research is conducted with the help of the students and faculty members of the Human Technology Interaction (HTI) research group at the Dept. of CSE, BUET, Bangladesh.

References

1. Lampe C, Ellison NB, Steinfield C (2008) Changes in use and perception of Facebook. In: Proceedings of the ACM conference on computer supported cooperative work, CSCW, pp 721–730
2. Hearst MA (2011) Natural search user interfaces, communications of the ACM, 54(11)
3. Key facts—Facebook newsroom (2013). http://newsroom.fb.com/Key-Facts. Accessed 31 Dec 2013
4. Alexa ranking of Facebook (2013). http://www.alexa.com/siteinfo/facebook.com. Accessed 31 Dec 2013
5. Facebook usage statistics (2013). http://www.thesocialskinny.com/100-social-media-statistics-for-2012/. Accessed 31 Dec 2013
6. Efron M, Winget M (2010) Questions are content: A taxonomy of questions in a microblogging environment. In: Proceedings of the 73rd ASIS&T annual meeting on navigating streams in an information ecosystem—volume 47, ASIS&T, 27:1–27:10
7. Teevan J, Ramage D, Morris MR (2011) #twittersearch: a comparison of microblog search and web search. In: Proceedings of the fourth ACM international conference on Web search and data mining, WSDM
8. Lampe C, Vitak J, Gray R, Ellison N (2012) Perceptions of Facebooks value as an information source. In: Proceedings of the SIGCHI conference on human factors in computing systems, CHI
9. Morris MR, Teevan J, Panovich K (2010) A comparison of information seeking using search engines and social networks. In: Proceedings of the international AAAI conference on weblogs and social media, ICWSM
10. White RW, Richardson M, Liu Y (2011) Effects of community size and contact rate in synchronous social q&a. In: Proceedings of the SIGCHI conference on human factors in computing systems, CHI, pp 2837–2846
11. Jeong J, Morris MR, Teevan J, Liebling D (2013) A crowd-powered socially embedded search engine. In: Proceedings of the international AAAI conference on weblogs and social media, ICWSM
12. Liu Z, Jansen BJ (2013) Analysis of factors influencing the response rate in social q&a behavior. In: Proceedings of the ACM conference on computer supported cooperative work, CSCW
13. Morris MR, Teevan J, Panovich K (2010) What do people ask their social networks, and why?: a survey study of status message q&a behavior. In: Proceedings of the SIGCHI conference on human factors in computing systems, CHI, pp 1739–1748
14. Panovich K, Miller R, Karger D (2012) Tie strength in question & answer on social network sites. In: Proceedings of the ACM conference on Computer supported cooperative work, CSCW, pp 1057–1066
15. Farnham S, Lahav M, Raskino D, Cheng L, Laird-McConnell T (2012) So.cl: An interest network for informal learning. In: Proceedings of the international AAAI conference on weblogs and social media, ICWSM
16. Naaman M, Boase J, Lai CH (2010) Is it really about me? message content in social awareness streams. In: Proceedings of the ACM conference on computer supported cooperative work, CSCW
17. Teevan J, Morris MR, Panovich K (2011) Factors affecting response quantity, quality, and speed for questions asked via social network status messages. In: Proceedings of the international AAAI conference on weblogs and social media, ICWSM
18. Hecht B, Teevan J, Morris MR, Liebling DJ (2012) Searchbuddies: bringing search engines into the conversation. In: Proceedings of the international AAAI conference on weblogs and social media, ICWSM
19. Horowitz D, Kamvar SD (2010) The anatomy of a large-scale social search engine. In: Proceedings of the 19th international conference on World wide web, WWW, pp 431–440

20. Yang J, Morris MR, Teevan J, Adamic LA, Ackerman MS (2011) Culture matters: a survey study of social q&a behavior. In: Proceedings of the international AAAI conference on weblogs and social media, ICWSM
21. United Nations E-Government Survey (2010). http://www2.unpan.org/egovkb/documents/2010/E_Gov_2010_Complete.pdf. Accessed 31 Dec 2013
22. Internet user statistics (2013). http://data.worldbank.org/indicator/IT.NET.USER/countries. Accessed 31 Dec 2013
23. Facebook user statistics (2013). http://www.socialbakers.com/facebook-statistics/page-2/?interval=last-6-months. Accessed 31 Dec 2013
24. Ferdous HS, Choudhury FM, Rifat MR, Moutushy S (2012) Usability analysis of e-Governance services in Bangladesh a survey and future directions. In: Prooceedings of the 15th international conference on computer and information technology (ICCIT)
25. Survey questionnaire on Q/A behavior in Facebook. https://docs.google.com/forms/d/1Px8hXj9NEz1SuO9skaBlmHKnUW9X13gkmHsrzXB4eQg/viewform?pli=1

Evolutionary Influence Maximization in Viral Marketing

Sanket Anil Naik and Qi Yu

Abstract With the growth of social networks, significant amount of data is brought online that can benefit applications of many kinds if being effectively utilized. As a typical example, Domnigos proposed the concept of viral marketing, which uses the "word of mouth" marketing technique over virtual networks (Domingos, IEEE Intell Syst 20:80–82, 2005). Each user is associated with a network value that represents his/her influence in the network. The network value is used along with other intrinsic features that represent user shopping behaviors for the selection of a small subset of most influential users in the network for marketing purpose. However, most existing viral marketing techniques ignore the dynamic nature of the virtual network where both the features and the relationship of users may change over time. In this paper, we develop a novel framework for the selection of users by exploiting the temporal dynamics of the network. Incorporating temporal dynamics of the network would assist in selecting an optimal subset of users with the maximum influence over the network. This paper focuses on developing an algorithm for the selection of the users to market the product by exploiting the temporal and the structural dynamics of the network. Extensive experimental results over real-world datasets clearly demonstrate the effectiveness of the proposed framework.

Keywords Viral marketing · Subset selection · Evolutionary · Network value · Influence flow

1 Introduction

The exponential growth of the Internet has transformed the Web into a *virtual world*. As most people in the real world have become a part of this virtual world, their social experience has also been translated into the Web. The increasing popularity over social network sites, such as Facebook, LiveJournal, and Twitter indicates

S.A. Naik (✉) · Q. Yu
Rochester Institute of Technology, Rochester, NY, USA
e-mail: san8774@rit.edu

Q. Yu
e-mail: qi.yu@rit.edu

Ö. Ulusoy et al. (eds.), *Recommendation and Search in Social Networks*,
Lecture Notes in Social Networks, DOI 10.1007/978-3-319-14379-8_11

the immense interactions of the users on the Web. The large-scale data resulted from social interactions over the Web forms a rich information repository that has the potential to benefit various applications. Marketing is a typical example of such applications. Traditional Web-based marketing mechanisms are more inclined toward direct marketing, which identifies the most probable customers and then markets the product or service directly to them. Although direct marketing ensures that marketing is delivered directly to potential customers, it is a slow and expensive process especially when targeting thousands of millions of online users. If the market cannot be conducted in a timely fashion, valuable business opportunities may get lost.

Different from direct marketing that treats each customer as an isolated entity, *viral marketing* regards users as part of a connected network and aims at selecting a subset of users in the network to market with an ability to influence other members of the network [1]. The interactions between users in the network help achieve this objective via implicit or explicit recommendation. Individual's actions also contribute toward influencing people around a user. For example, people tend to look at what others around them are using or buying. A person who has a better understanding of the preference of their friends is more likely to make proper recommendation on products or services. These behavioral phenomena will further strengthen the effectiveness of viral marketing. Some recent statistics of social network user behavior provide clear evidence to justify the significant potential of viral marketing. For example, among all users of the major social network sites (e.g., Facebook, Twitter, and MySpace), 20 % of them share content of the network using the share option [2]. A 2009 research reveals that 32 % of the users share promotional offers inside a private social network [3] whereas 51 % of the users click "forward to a friend" in marketing emails [4].

In social networks, the ability of a user to influence others increases with the number of connections or interactions with other users in the network. Hence, viral marketing when implemented properly can grow exponentially. This exponential growth can be simplified and represented in the form of a pyramid, where users at every level are influenced by the users in the above level and have the responsibility to influence the users in the level below. Figure 1 represents the influence of viral marketing in the pyramid form where each user at every level influences two users in the level below. Hence, the number of users influenced at a particular level can be determined by $2^{\text{level}-1}$ and the total number of users influenced by a single user may be as large as $2^{\text{level}} - 1$.

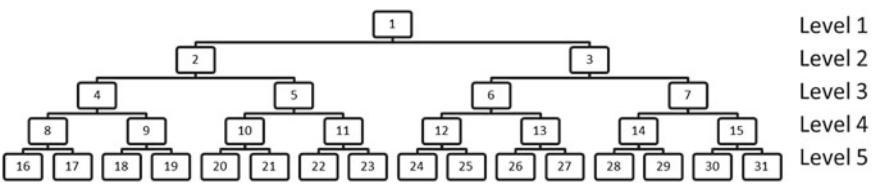

Fig. 1 Pyramid flow in viral marketing

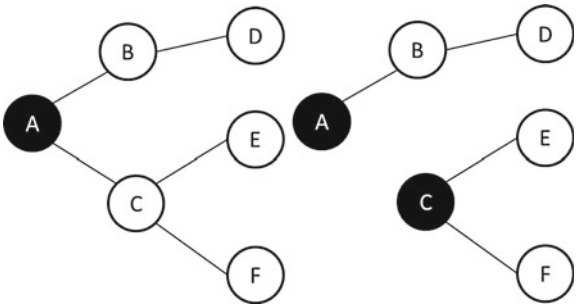

Fig. 2 Network graphs at time t_1 and t_2

A key limitation with existing viral market solutions is that they neglect the dynamic nature of the social relationships. As the social relationship changes over time, the relationship between users on the Web also changes. In social networks, new users register and existing users deactivate their accounts over time. The relationship between users also changes frequently. For example, a user A may be unknown to user B at time t_i, but may become good friend in time t_j for some $j > i$. Furthermore, the attributes of users can also change over time. For example, a user X's marital status might change between two time windows. The effectiveness of the transmission of knowledge depends upon recording these changes and adjusting accordingly to adapt to these changes. For example, the left graph in Fig. 2 representing a part of the social network at time t_1. Since there is a path to reach E and D from A, it is possible to influence E and D by influencing A. However, at time t_2, the network has a changed structure represented by the right graph in Fig. 2. In the new structure, the link between A and C is lost thereby making it impossible to influencing E from A and thus requiring C to be influenced separately at time stamp t_2. Thus, it is necessary to adapt to the changes in a dynamic network to maximize the influence of marketing.

In this paper, we propose a novel framework to effectively apply viral marketing in a dynamic social network.

2 Preliminaries

The selection of users in viral marketing is based on the concept of *network value*, where a user with a higher network value has more influence over other users in the network. The network value is determined by two key factors: the *intrinsic value* of a user and the *connectivity value* of the network. In this section, we detail these two important factors and then describe some other relevant concepts that are used throughout the paper.

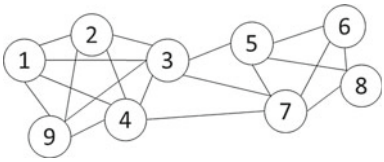

Fig. 3 Dense subgroups in the network

The intrinsic value is a normalized score calculated as a composition of various attributes of a user, which include things like recommendations made in a social network (e.g., Facebook), messages forwarded in an email system (e.g., Gmail), and so on. Some specific requirement of products or services may also be helpful to determine the intrinsic score. For example, the marketing of baby products may include the attribute of marital status, while marketing of ladies' perfumes may include the gender attribute. The activeness of the users in network also adds toward the calculation of the intrinsic value. The activeness in a social network could be calculated as the function of number of posts posted per day.

While the intrinsic value measures users' individual attributes, the connectivity value measures the network structure as a whole. It is a function of not only how well the user is connected in the network but also how well his/her neighbors are connected in the network. To effectively spread the influence in a network, it is necessary to hit the network from different ends, which is similar to the spread of an epidemic. The overall network can be usually regarded as a composition of a number of smaller strongly intraconnected subnetworks. Identification of such subnetworks and targeting users from each subnetwork is essential for fast spread of influence. As an example, in Fig. 3, nodes 1, 2, 3, 4 and 9 are strongly intraconnected while nodes 5, 6, 7, and 8 are strongly intraconnected, thereby forming two subnetworks, which are weakly interconnected. To effectively spread the influence, it would be necessary to select nodes from both subnetworks.

Another key concept used by the paper is *influence flow*, which is represented by a function of the live edges directed toward a user. An edge is regarded as live if its source is influenced. Thus, the probability of a user to get influenced in a particular time step is proportional to the number of influenced neighbors. Since the social network is inherently dynamic, we use subscript t to denote the temporal dynamics. Let S_t denotes the subset of users selected from the graph G_t at time step t. When there is a change on the network graph from $t-1$ to t, S_t should be changed accordingly. However, it is important that S_t does not deviate too much from the recent past due to a sudden change in a given time step. This actually is a reasonable expectation as a sudden change should mostly be due to an existence of some noise (e.g., a user accidentally removes a friend), which may be fairly common in a highly dynamic social network environment. Hence, the temporal knowledge obtained between the time windows provides an evolutionary outlook to system where it remains faithful to the current time window and not deviates significantly from the history [5].

3 The Evolutionary User Selection Framework

This section describes the framework for the evolutionary selection of users to maximize the viral marketing influence. The initial step for the selection of influencers in the network is to determine their network values, which aggregate the intrinsic values of individual users and the connectivity value of the network. Then a number of smaller strongly intraconnected subnetworks are identified to enable the selection of the users in different parts of the network. We adopt a threshold-based approach to compute the influence flow in the network, where an inactive user (not influenced) gets influenced by an active user (influenced) if the number of direct active friends of that user goes beyond the threshold [6]. We introduce a novel evolutionary metric to monitor the influenced users after every time window to determine the need for the additional selection of users with respect to the change in the graph in that time window.

3.1 Network Value Calculation

The network value measures a user's capacity as an influencer in the network rather than a customer. The network value is determined as a function of intrinsic value and connectivity, where the former is the composition of various attributes related to the product or service to market along with other relevant data available from the network and the latter represents how well a user is connected with other users in the network.

3.1.1 Intrinsic Value

Intrinsic value (I) represents how well the user can be associated with a particular product or service as an influencer for marketing. The base calculation requires the computation of Feature Mapping and Recommendation Score.

Feature Mapping (M) is used to determine if a particular user can act as an influencer for a particular product or service. To determine this, every product or service is represented by a set of attributes and a set of normalized values $\{Nv_k\}$ are used to represent the strength of the features for that user, where normalization is used to scale all the required features to the same level. Weights $\{W_k\}$ are provided to give different importance to different features as required. More specifically, the feature mapping M of a user is defined as

$$M = \sum_k W_k \times Nv_k \tag{1}$$

Recommendation score (R) represents if a user can be viewed as a good recommender. The recommendation score is calculated based on not only the capability of the user itself but also his or her friends' capability of forwarding recommendations to others. Intuitively, a user with a highly influential friend tends to be influential as well. Specifically, the recommendation score R of a user is defined as

$$R = \frac{n}{N} \times \sum_{k=1}^{n} \frac{r_k}{Tr_k} \tag{2}$$

where n denotes the number of friends that receive recommends from the user, N is the total number of friends of the user, r_k is the recommendations forwarded by the kth friend, and Tr_k is the total recommendations received by the kth friend.

Finally, the *Intrinsic value* I_i of the ith user is calculated as

$$I_i = (W_M \times M_i) + (W_R \times R_i) \tag{3}$$

where W_M and W_R are weights against feature mapping and recommendation score of the user, respectively.

3.1.2 Connectivity Value

Connectivity value (C) represents not only on how well a user is connected in the network but also how his/her friends are connected in the network. It is necessary to select users whose friends are also well connected in the network to ensure the flow of influence beyond the secondary level. This is essential to achieve an exponential growth as illustrated in Sect. 1. Hence, we compute the connectivity value C_i of the ith user as

$$C_i = \frac{\sum_{k \in S_i} |S_k|}{|S_i|} \tag{4}$$

where S_i is the set of friends of the ith user.

3.1.3 Network Value

Network Value (Nv_i) of the ith user is computed by aggregating its intrinsic value and connectivity value. The intrinsic value and the connectivity first normalized before being aggregated.

$$I_i = \frac{I_i - I_{\min}}{I_{\max} - I_{\min}} \quad C_i = \frac{C_i - C_{\min}}{C_{\max} - C_{\min}} \tag{5}$$

$$Nv_i = (W_I \times I_i) + (W_C \times C_i) \tag{6}$$

where I_i is the normalized intrinsic value, C_i is the normalized connectivity value, W_I and W_C are weights against intrinsic and connectivity values, respectively.

3.2 Relationship

Relationships between the users in a network are represented by the edges connecting those users. Presence of an edge represents the existence of a relation. The strength of the relation is specified by the edge weight. A positive value shows a relation favorable for the flow of influence. The calculation of this strength is based on the information available from the network.

Communication weight (Wc) specifies the strength of the relation based on the amount of communication between the users. It is determined by the sum of the communication between the two users.

$$Wc_{ij} = \sum c_{ij} \tag{7}$$

where c_{ij} represents each communication sent from user i to user j.

Recommendation weight (Wr) represents the trust between the users. The trust can be positive or negative if the source user recommended or did not recommend the other user, respectively. This is a very strong relationship factor and also provides additional knowledge in semi-supervised grouping.

$$Wr_{ij} = \sum r_{ij} \tag{8}$$

where r_{ij} represents each recommendation/nonrecommendation of user j by user i.

The weight (W_{ij}) of the relation/influence from user i to user j can be calculated as follows:

$$W_{ij} = (W_1 \times Wc_{ij}) + (W_2 \times Wr_{ij}) \tag{9}$$

where W_1 and W_2 are weights against communication weight and recommendation weight between the users, respectively.

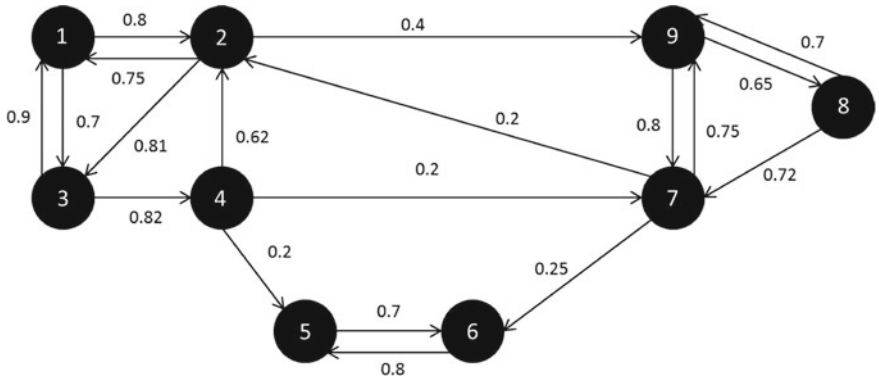

Fig. 4 A connected network example

3.3 Group Identification

A network as a whole can be represented as a composition of multiple dense subgraphs. These implicit groups have a strong association within the group and a lesser association between the groups. Dividing the network into such groups can be attained using graph-partitioning-based clustering algorithms, which identify groups of nodes that are strongly intraconnected and weakly interconnected. The intraconnectivity strength of a group bolsters the spread of influence in that group. Selecting influential candidates from each group allows attacking the network from different ends. This approach allows a faster spread of influence by dividing the network into smaller connected subnetworks.

Classical graph-partitioning-based clustering algorithms typically involve the calculation of the Eigen-decomposition of a graph [7], which has a cost of $O(N^3)$, where N is the total number of nodes in the graph [8]. Hence, it is computationally infeasible to directly apply these algorithms to large-scale social graphs. As our goal is not to precisely identify the clusters of nodes, we propose a fast graph partition algorithm for group identification based on the concept of connected components. The classical connected component algorithm uses a breadth-first (or depth-first) search algorithm to identify the connected subnetworks in a larger network [9]. However, this does not directly fulfill our requirement of identifying strongly connected groups or subnetworks in a network. For example, Fig. 4 shows a connected network. Directly applying the connected component algorithm on this network results in a single network as all the nodes in the network are connected.

To resolve this issue, we make the following modification of the classical algorithm. Specifically, before running the algorithm, we create an abstract view of the existing network by ignoring the weak edges. The resulting view is a network (V, E') where V represents the nodes in the original network and $E' \subset E$, i.e., subset of strong edges from the original set E. We use the box plot [10] approach to achieve outlier robustness when determining the threshold to assign edges into strong or

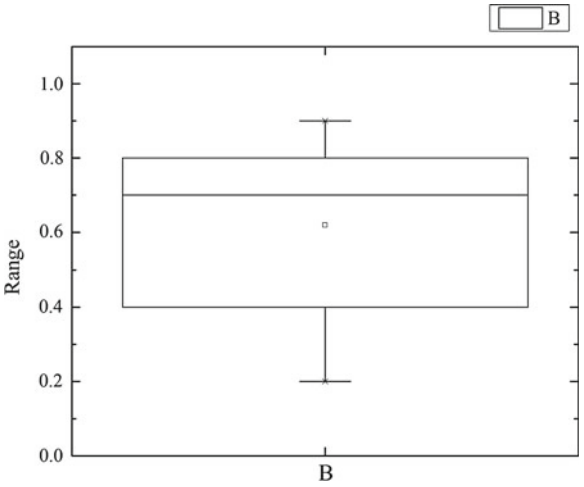

Fig. 5 Box plot of edge weights

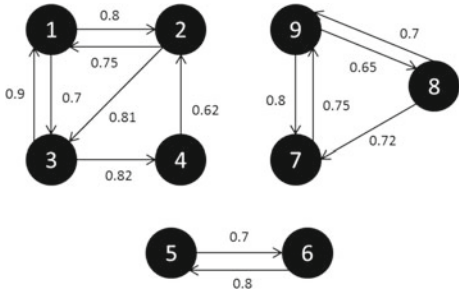

Fig. 6 Abstract view of the network

weak categories. As shown by the box plot in Fig. 5, the example network has a mean edge weight of 0.62. Ignoring edges below the mean results in an abstracted view shown in Fig. 6. Applying the connected component algorithm on this abstract view gives three distinct subnetworks. The nodes in each subnetwork are connected by strong edges, which represent a potential to direct an influence flow within that network. The graph traversal principle guarantees the time complexity to be linear in the number of nodes of the graph and hence achieves performance with orders of magnitude better than graph-partitioning clustering algorithms. The connected component philosophy also assures the identified groups to be internally connected.

Prior knowledge of users may also be leveraged to enhance the grouping process. For example, it is necessary to group users with close associations together. Close associations can be represented by coauthorship in the authorship dataset, trust–distrust in trust-based dataset, and followership in the micro blogging dataset. This

prior knowledge might not always be conspicuously represented in the network. We adopt an award-penalty mechanism to incorporate the prior knowledge into the group identification process:

$$E'_{ij} = E_{ij} + R_{ij} W_{\text{reward}} - P_{ij} W_{\text{penalty}} \tag{10}$$

where E_{ij} denotes the original weight of the edge, $R_{ij} \in \{0, 1\}$ represents whether E_{ij} is a must link (i.e., i and j should be grouped together), $P_{ij} \in \{0, 1\}$ represents whether E_{ij} is a can-not link (i.e., i and j should not be grouped together), W_{reward} is the reward weight, and W_{penalty} is the penalty weight.

3.4 Influence Flow

The flow of the influence in the network is directed in accord to the threshold-based model specified by Kleinberg et al. [6]. A user is influenced if its influence value is higher than the threshold specified. This influence value (Iv) is a function of the number of influenced neighbors and the strength of the relation.

$$Iv = f(Ni, \sum_{k \in S_{\text{ie}}} W_k) \tag{11}$$

where Ni represents the total number of influenced neighbors, W_k denotes weight of the k^{th} edge, and S_{ie} represents the set of edges from influenced neighbors.

This function shows the ability of the user to get influenced based on the current state of the network. This function is represented by two functions viz. f_p and f_{s}. f_p is the percentile function of the *live edges* and f_{s} is the function of the total live edge strength with respect to the network.

An incoming edge is termed as a *live edge* if its source is an influenced user and the destination is not. The function f_p represents the percentile strength of the live edges for a user, i.e., the influence of the live edges with respect to the other incoming edges for a user

$$f_p = \frac{\sum_{k \in S_{\text{ie}}} W_k}{\sum_{k \in S_e} W_k} \tag{12}$$

where S_{ie} represents the live edge set. S_e denotes the set of all edges for the user, i.e., $S_{\text{ie}} \subset S_e$. W_k represents the weight of the k^{th} edge.

The value of f_p is compared with the threshold value θ set in the model. The selection of θ is based on empirical analysis of the dataset conforming closest to the natural flow of influence. The other function f_{s} represents the strength of live edges.

$$f_{\mathrm{s}} = \sum_{k \in S_{\mathrm{ie}}} \tag{13}$$

The value from f_{s} is compared with the average incoming edge strength in the network ($\mathrm{Te}_{\mathrm{avg}}$)

$$\mathrm{Te}_{\mathrm{avg}} = \mathrm{Me} \times \frac{\mathrm{Te}}{\mathrm{Tn}} \tag{14}$$

where Me is the current median edge weight in the network, Te represents the total edges in the network, and Tn denotes the total nodes in the network.

$$f_{\mathrm{s}} > \theta \text{ and } f_p > \mathrm{Te}_{\mathrm{avg}} \tag{15}$$

The impetus in using these two threshold measures is to give comparable opportunity for each user to be influenced. The function f_p mathematically favors the users with less number of neighbors, while function f_{s} favors one with large number of neighbors. The logical combination of these two functions in Eq. (15) provides a balance to the flow of influence.

3.5 Dealing with Dynamic Changes of the Network

The social network is not static. There are continuous changes like the addition of new users or deletion of the old ones. These changes include not only the change of relationship between users but also the attribute change of individual users. Nevertheless, these important changes go unrecorded in static graphs, which may significantly affect accuracy of user selection in viral marketing.

As the social network environment is highly dynamic and complete autonomous, changes should be treated as norms instead of exceptions. Meanwhile, many changes could be introduced due to various noises (e.g., accidentally adding or removing a friend). Hence, the framework should be robust enough to overcome the noises while being able to adapt to the changes. To achieve this dual objective, we propose to follow the temporal smoothness principle to cope with changes in the network, which demands the selection of users to respect the current snapshot of the network while not deviating dramatically from the recent past.

$$Nv = (1 - \alpha) \times Nv_{\mathrm{historical}} + \alpha \times Nv_{\mathrm{current}} \tag{16}$$

$$Wij = (1 - \alpha) \times Wij_{\mathrm{historical}} + \alpha \times Wij_{\mathrm{current}} \tag{17}$$

Adapting to this dynamic nature of the network is facilitated by growth rate (g). Growth rate measured at every time step represents the strength of the influence in the network. This strength is represented as

$$g_t = \frac{Vi_t}{Vu_t} \tag{18}$$

where g_t denotes the growth rate in time step t, Vi_t represents the influenced users in t, and $Vu_t = V - Vi_t$ representing the noninfluenced users in t with V specifying the total number of users in the network.

At each time window with the change in the network structure a new provisional growth rate is calculated against the new potential set of users for the next iteration

$$g'_t = \frac{Vi'_t}{(Vu'_t - Vi'_t)} \tag{19}$$

where g'_t represents the provisional growth rate at time step t with Vi'_t representing the potential list of influenced users and Vu'_t is the total number of noninfluenced users after t.

The potential list is based on the current visible users in the network with the network value (Nv) near threshold. If $g'_t < g_t$, then the growth rate would potentially decline in the next window. To counter this we calculate the number of users required to maintain the growth rate g_t.

$$Vd = (g_t \times (Vu'_t - Vi' - t)) - Vi'_t \tag{20}$$

where Vd is the additional number of users required.

We directly market the Vd best possible users from the network exclusive of Vi'_t to maintain the growth rate. This adaptability ensures the flow of influence maintained in the continually changing network.

3.6 The Algorithm

The algorithm of the proposed model can be broken down into three distinct blocks viz. *initialization*, *temporal update*, and *influence flow*. Algorithm 1 represents the evolutionary marketing function. This function selects the call to initialize or update the network based on time window. Algorithm 2 denotes the initialization block. This block is called when there is no prior network information or the previous network information needs to be overwritten with the current network. An existing network is updated using Algorithm 3. The flow of influence is reevaluated using block 4. The flow of influence in the network is described by Algorithm 5.

Algorithm 1 Evolutionary User Selection

```
  function EVOLUTIONARY(G′, w)
Input: The Time window w
Input: The Network Graph update G′
    if w = 1 then
      G = G′
      call function InitializeInfluence(G)
    else
      G = UpdateNetwork(G,G')
      call function EvolutionaryInfluence(G,w)
    end if
  end function
```

Algorithm 2 Initialize_Influence

```
Input: The network graph G
  D = φ                                              ▷ direct market list
  C = IdentifyGroups(G)
  for each c ∈ C do
    n = # top users to select from c
    if n > 0 then
      while i < n do
        v = getTop(c, i)                             ▷ extract ith ranked v
        if v is not influenced then
          D = D ∪ v
          if state of v not visited then
            set v state to visiting
          end if
        end if
        i ← i + 1
        if i > |c| then
          break
        end if
      end while
    end if
  end for
  V = D                                              ▷ Initialize the visiting list
  for each node v ∈ D do
    set v as influenced
    if state of v is visiting then
      add neighbors of v to V
    end if
  end for
  update the potential list P with V using eq (15)
  call function InfluenceFlow
```

Algorithm 3 Update_Network

```
function UPDATENETWORK(G, G′)
Input: The network Graph G(V, E)
Input: The network Graph update G′(V′, E′)
Output: The updated network Graph G
    for each v ∈ V∪V′ do
        if v ∈ V & v ∈ V′ then
            update Nv using eq (16)
        else if v ∈ V then
            update Nv with Nv_current = 0 using eq (16)
        else if v ∈ V′ then
            add v to V
        end if
    end for
    for each e_ij ∈ E∪E′ do
        if e_ij ∈ E & e_ij ∈ E′ then
            update Wij using eq (17)
        else if e_ij ∈ E then
            update Wij with Wij_current = 0 using eq (17)
        else if e_ij ∈ E′ then
            add e_ij to E
        end if
    end for
  return G
  end function
```

Algorithm 4 Update_Influence

```
function EVOLUTIONARYINFLUENCE(G,w)
Input: The network Graph G
Input: The time window w
    update the potential list P eq (15)
    calculate potential growth rate g′ eq (19)
    if g′ < g then
        calculate additional users Vd using eq (20)
        directly market Vd
    end if
    call function InfluenceFlow
  end function
```

Algorithm 5 Influence_Flow

```
function INFLUENCEFLOW
    V = φ                                                    ▷ visiting list
    P = φ                                                    ▷ Potential list
    T = φ                                                    ▷ temp visiting list
    while (size(V) > 0 || size(P) > 0) & (nW ≠ true) do      ▷ nW =new window availability
        for each v ∈ P do
            if f_s > threshold & f_p > threshold then
                set v as influenced
            end if
        end for
        update growth rate g using eq (18)
        for each v ∈ V do
            if state of v not visited then
                add v neighbor to T
            end if
        end for
        append V with T
        update P with V using eq (15)
    end while
end function
```

4 Experiments

We conduct a set of experiments to assess the effectiveness of the proposed evolutionary user selection framework for viral marketing. For comparative purpose, we include two nonevolutionary user selection approaches. More specifically, the first approach, referred to as *Non Evol Group*, uses the group identification feature, which allows it to attack the network from different ends by selecting most influential candidates from each group. The second approach, referred to as *Non Evol No Group*, selects the most influential candidates from the entire network independent of their group presence. Following the same naming convention, we refer to the proposed approach as *Evol Group*. The key metrics we evaluate include the total influences in the network and the ability to sustain the flow of influence in the network.

4.1 Dataset and Experiment Setup

The experiments are conducted over a real-world dataset collected from large-scale social network. As the data is collected over a long period of time, the temporal dynamics are clearly captured by the dataset.

4.1.1 FriendFeed

The dataset that we have used is a well-known micro blogging and social network service "FriendFeed" (http://friendfeed.com). The microblogging feature in association with user's ability to follow a particular *entry* (like in well-known Twitter) or "like" or "comment" on one (like Facebook) provides a vast pool of social data [11]. The structure of the dataset available comprised of followers, entries, comments, likes, users, and networks for the date monitored between August 1, 2010 and September 30, 2010. Mining the information based on the contents of the entries and comments is ignored as it is out of scope for current research work. Current research monitors the quantitative interaction between the users on the micro blogging service. The users are represented as the nodes of our graph.

Directed edge is present from user B to user A if

A follows B or
A comments on an entry by B or
A likes an entry of B

These interactions represent the ability of the user B to influence user A directly or indirectly. The data was processed to ignore orphaned entries and users with no interactions in the network for our experimental needs. The resulting network was represented by 22,817 nodes and 303,785 edges. For evolutionary analysis the dataset between August 1, 2010 and September 30, 2010 was divided into 12 windows, thus each window representing data for 5 days.

Network Value Calculation: Friendfeed provides extensive information on the social activity of a user. The quantitative information like number of followers, total entries in the social network, comments made and received, likes made and received for the entries constitute toward the network value of the user in our graph. These quantitative measures are represented in a normalized format and incorporated with the associated weights. This measure provides intuitive information on the activeness of the user in the network and the overall influence they represent.

Edge Weight Calculation: The edges in our graph exhibit the social relationship between the users in the network. The number of comments and likes shared between the users along with the follower information is used to represent this relationship. The edges representing fellowship are identified as *must-link edges*. In parallel to HEP-Th these *must-link* edges are used in the semi-supervised grouping.

The evolutionary model reads each window at a time in confidence of the evolutionary principles, whereas the nonevolutionary models look at a single snapshot of aggregated data, there by losing the temporal knowledge. The evolutionary algorithm updates the graph after each window by conforming to the evolutionary concept of maintaining the low history cost and representing high snapshot quality [5]. The nonevolutionary-based algorithms are allowed to run till they converge, i.e., either *influence* the entire network or no more *influence* update available. Keeping with the scope of this paper, for simplicity the algorithms are allowed to run at three equally spaced out threshold levels ($\theta = 0.25, \theta = 0.5, \theta = 0.75$).

Figure 7 shows the total number of users influenced by each model at different threshold levels θ and Fig. 8 compares the number of iterations the flow persisted in the network. Both graphs show the supremacy of the evolutionary model over the nonevolutionary-based model. The evolutionary model influences far more users than its nonevolutionary counterpart and runs for large number of iterations. This high performance can be attributed to the capture of the temporal change in the network missed by the nonevolutionary-based models that looked at static aggregates' snapshot. The large number of iterations provides the confidence of a longer run of *influence* as well as larger coverage of the influence in the network.

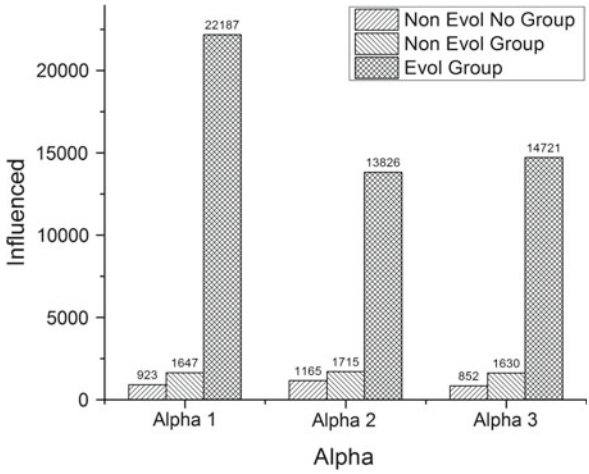

Fig. 7 FriendFeed-influenced users

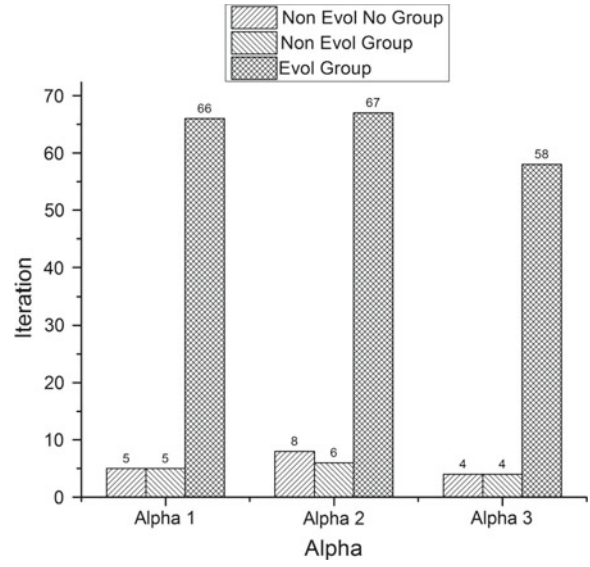

Fig. 8 FriendFeed-iterations

This ability to run for a longer period ensures the *influence* flow kept alive which is evident from the growth rate representation in Fig. 9. It can be observed that the growth rate for the nonevolutionary model represents a single spike. The growth rate increases till it discovers the network from the initial selection whereby after reaching the saturation level the growth rate steadily decreases. In contrast, the growth rate for the evolutionary model is represented by multiple spikes. The multiple spike results from the continual learning of the network as the new data comes in at each time window and self-adjusting the growth rate to keep the *influence* flowing in the network, thereby restoring the confidence of the *influence* flow in the network. The inability to grow from initial selection could be the result of loss of relational information in the aggregated graph and improper selection of the users. Figure 10 demonstrates a similar analysis for the total users' influences against the iterations. The nonevolutionary models grow quickly and reach the saturation level, whereas the evolutionary model steadily increases in step mode. The step level represents the knowledge of new data flowing-in for a window. This suggests that the supremacy of the proposed evolutionary algorithm in influencing the users in coauthorship network with an assurance of maintaining the flow of *influence* by adjusting to the incoming data.

The supremacy of the evolutionary model, specifically in this data, can be contributed to the fact that the data is highly volatile as expected from any social network. This temporal volatility is successfully captured by evolutionary model as it by-par outperforms the nonevolutionary model.

4.1.2 HEP-Th

The second dataset we use in our experiments is the HEP-Th dataset. HEP-Th dataset represents the information on papers in theoretical high-energy physics from arXiv (www.arxiv.org). The collaboration graph represents the relationship between journals, papers, and authors as represented in Fig. 11. The structure of this dataset provides a key feature of coauthorship in a social network. The data captures the relationship between the authors. The authors are represented as the nodes of network graph. An edge is created from author B to author A if (1) A refers a paper by author B or (2) A coauthors a paper with author B. These interactions represent the ability of the author B to influence author A directly or indirectly. The data was preprocessed to remove missing data and corresponding authors with no connection to the network were ignored. Authors of the papers not present in the author list were added with synthetic identifiers. The resulting graph is a close approximation of the original graph. Amongst the 49 distinct research areas available in the dataset the experiments were run only with reference to High-Energy Physics, Atomic Physics, History of Physics, Computational Physics, Numerical Analysis, and Classical Physics.

Network Value Calculation: The network value of the author in this network is calculated with reference to (1) Number of papers in the targeted area of influence, (2) Number of citations received, (3) Total number of downloads in the first 60 days,

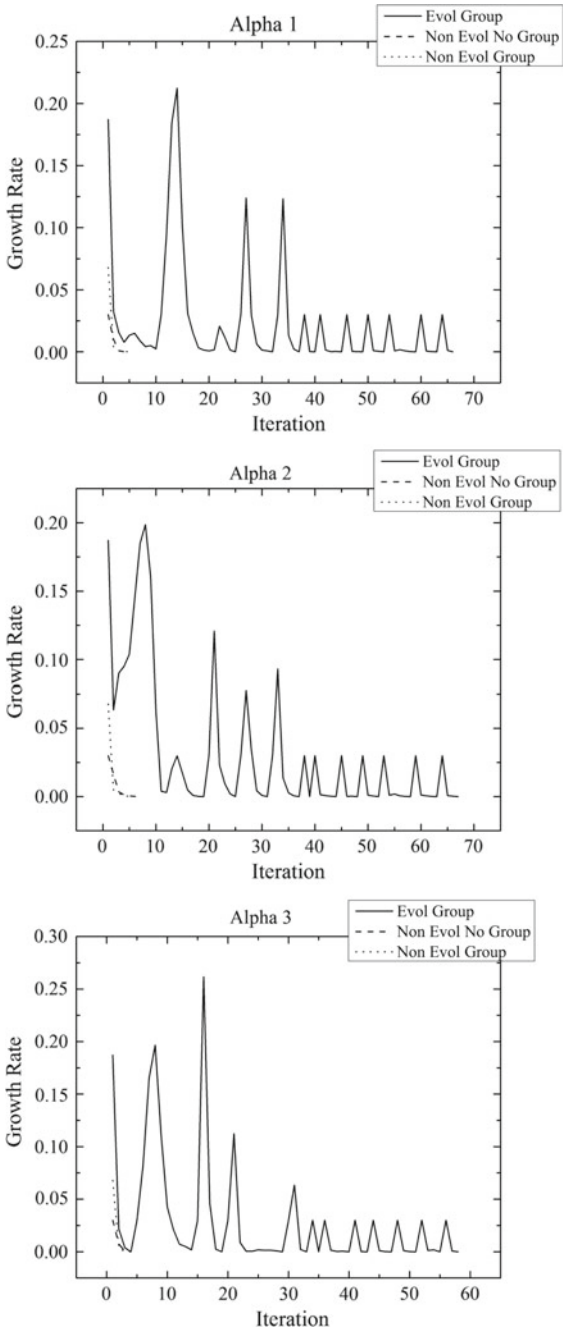

Fig. 9 FriendFeed-growth rate against iteration at each θ

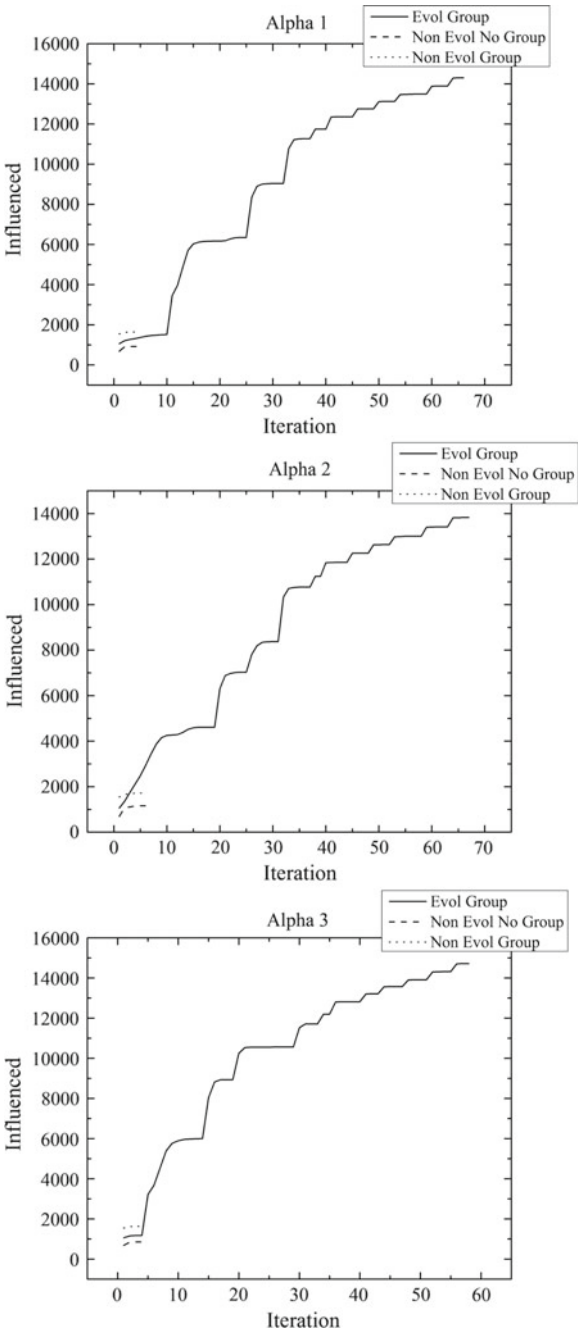

Fig. 10 FriendFeed-influenced against iteration at each θ

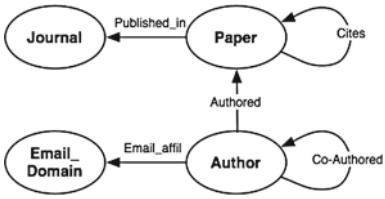

Fig. 11 HEP-Th schema [12]

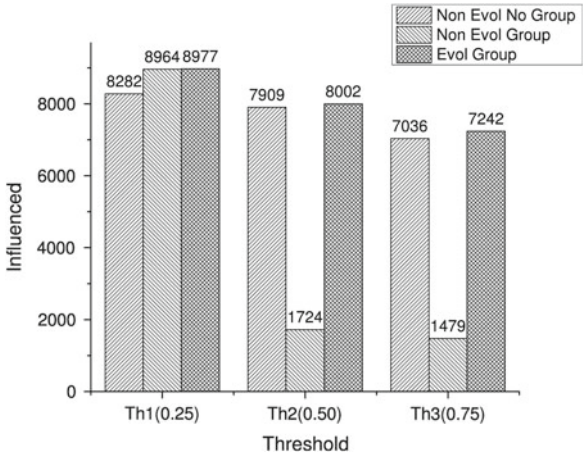

Fig. 12 # Influenced users versus threshold θ

and (4) Total number of papers published. Each factor is normalized based on the overall network and combined with the corresponding weight.

Edge Weight Calculation: The edges in our graph exhibit the relationships between authors. The weight of the edges between the nodes in this graph is presented as a combination of the coauthorship count between the two authors and cites reference count of node A to node B for an edge directed from node B to node A. The edges representing coauthorship is termed as *must-link* edges. *Must-link* edges are used for semi-supervised grouping. Similar to the network value calculation, the factors contributing toward the edge weight are normalized based on the network and combined with the associated weight

Figure 12 shows that the evolutionary algorithm outperforms both the nonevolutionary algorithms with respect to the total number of users *influenced* at all threshold levels. The difference in the total users influenced by the group-based nonevolutionary can be directed toward the alteration of the flow based on the change in the *influence* threshold level as pointed out earlier. Figure 13 demonstrates that the nonevolutionary model runs over a longer period of time as it learns about the new data after each window, thereby providing the confidence of a longer run of *influence*. Figure 14 shows that the growth rate for the nonevolutionary model represents a single spike. The growth rate increases till it discovers the network from the initial selection whereby after reaching the saturation level the growth rate steadily decreases. In contrast, the growth rate for the evolutionary model is represented by

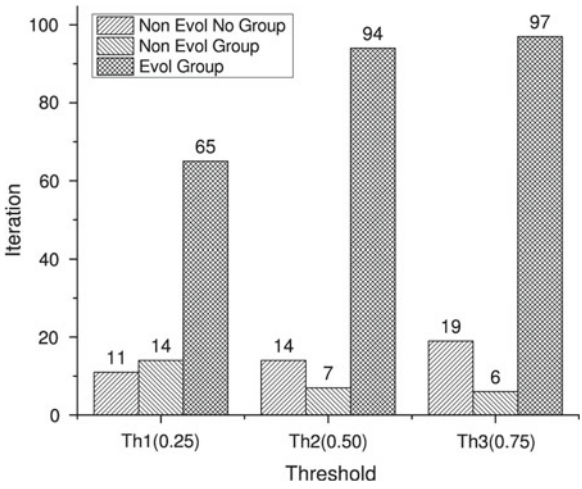

Fig. 13 # Iterations versus threshold θ

multiple spikes. The multiple spike results from the continual learning of the network as the new data comes in at each time window and self-adjusting the growth rate to keep the *influence* flowing in the network, thereby restoring the confidence of the *influence* flow in the network. Figure 15 demonstrates a similar analysis for the total users' influences against the iterations. The marginal gain in total influenced users of evolutionary over nonevolutionary model can be contributed toward the slowly changing nature of the network.

4.1.3 Epinion

Epinion is the third and final dataset that we used for our experimental analysis. It is one of the best known knowledge-sharing sites and termed as a "web of trust" for its trust relationship-based network. It allows users to post reviews in addition to rating. Users interact with each other by rating reviews and also by listing reviewers they trust. The "web of trust" employs the service to present reviews from trusted users first [13]. The architecture of the service provides information on authorship of articles, trust/distrust information, and product ratings by users which can be invaluable for any social network analysis model. For our experiment we captured the data on this service from January 2001 to September 2003. For evolutionary model this information was split into buckets of quarters thereby providing 11 windows for our analysis.

The data was processed to capture the information relevant to the relationship between the users in the network. The users are represented as the nodes in our corresponding graph representation with the edges symbolizing the relationship between the users. We represent a strong edge from user A to user B, if user A lies in the trust list of user B. An edge lies from user A to user B, if user B provides a positive

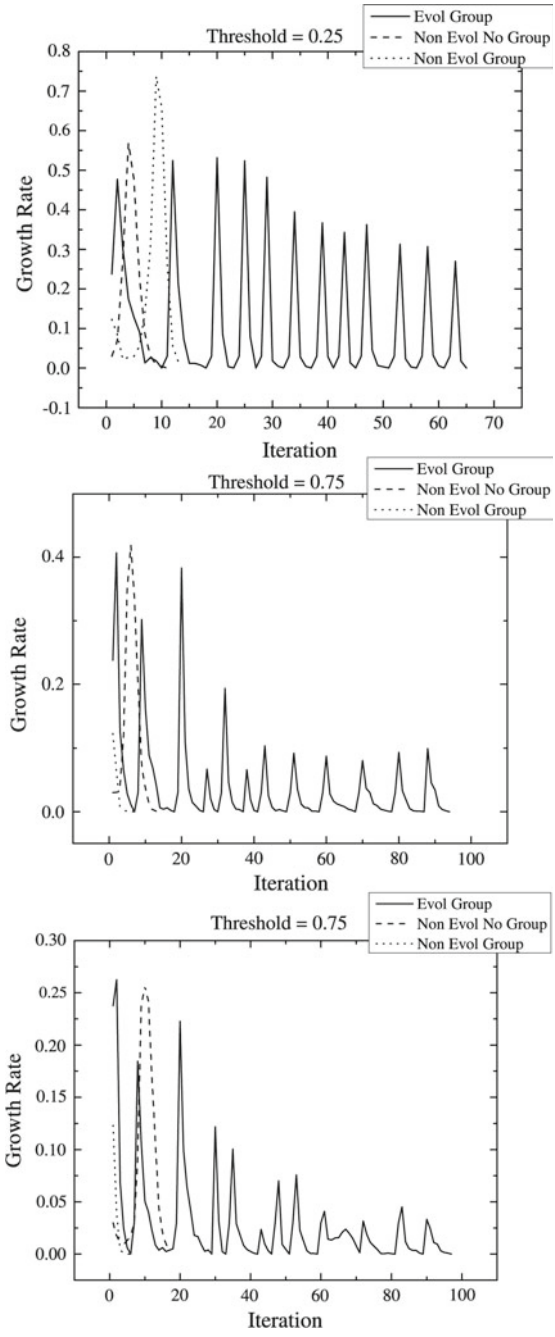

Fig. 14 Growth rate versus threshold θ

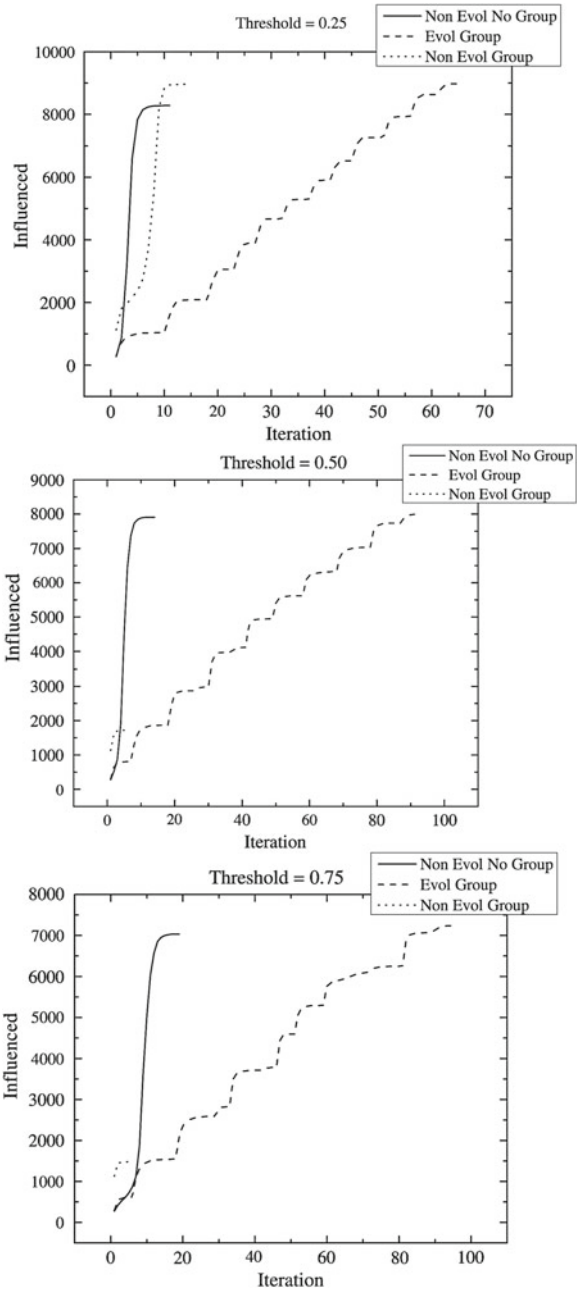

Fig. 15 # Influenced user versus threshold θ

rating to A's review. A negated edge lies if user B provides a negative rating to A's review. This approximate representation of the Epinion service resulted in 158,142 users and 5,053,088 edges.

Network Value Calculation: Epinion signifies trust-based influence information. This information can be perceived to determine the influence level of an Epinion user. Knowledge of trust/distrust count along with overall ratings and article counts is utilized to determine the network value of the user. These characteristics are normalized at the network level and combined with their corresponding weights.

Edge Weight Calculation: The edges provide a trust-based relation. Total ratings plus the trust/distrust information are used in the calculation of the edge weight. An edge representing a higher trust level is termed as *must-link* edge and one with higher distrust level is termed as *can-not link* edge. Semi-supervised grouping exploits this additional information.

Evolutionary model performs better than its comparative model in the total number of users influenced in different threshold levels as shown in Fig. 16. However, the nonevolutionary group-based model also performs significantly better than the greedy model. Approximately 97,000 and 85,000 more users are influenced by evolutionary and nonevolutionary group-based model in contrast to the greedy model. This huge difference between group- and nongroup-based model can be based on two possible reasons: (1) The influential nodes being concentrated at one end of the network thereby obstructing the opportunity to grow, (2) The graph being highly disconnected thereby reducing the possibility to reach different parts of the network. The group-based model works well against such cases. Figure 17 demonstrates that the total number of iterations used by evolutionary is higher as compared to the other models in line with the earlier observations. Growth rate exhibits the same behavior

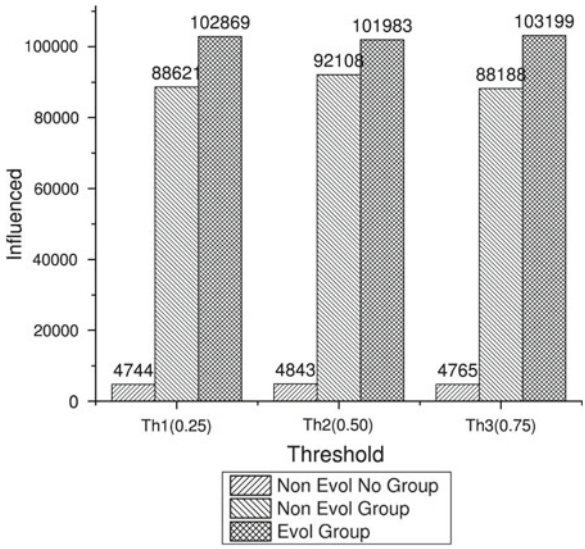

Fig. 16 # Influenced users versus threshold θ

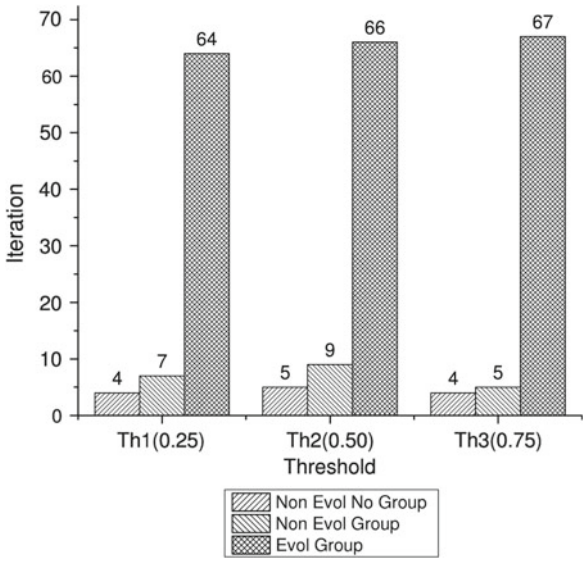

Fig. 17 # Iterations versus threshold θ

as FriendFeed for all the models as shown in Fig. 18. Figure 19 represents the total number of users influenced after each iteration. Likewise information on the early saturation of the nonevolutionary-based model and step-based growth of the evolutionary model can be inferred. However, another interesting aspect to capture here is the large number of users initially influenced for the group-based model as compared to the greedy model. This is only possible if the network is highly disconnected, which results in large number of groups and hence the large initial subset.

5 Related Work

The traditional marketing looks at customer as an individual and not as a part of the society with an ability to influence others. Viral marketing uses the network value of the users in the network in contrast to the customer value used by the direct marketing concept [1]. Domingos and Richardson [14] described network value as the composition of the connectivity of the user in the network and the users ability to influence other users in the network. The selection of the users can also be supported by the concept of predictive rating. Domingos and Richardson [15] proposed probabilistic approach similar to predictive rating determination. The model analyzes similar users liking and actions to determine the feasibility of marketing to a particular user in analogous to predict rating of the user from the analysis of similar users.

The virtual network for the viral marketing analysis can be represented by graph with nodes representing the users and edges representing the relationship between the users. Transaction logs and the event lists provide a large amount of data for

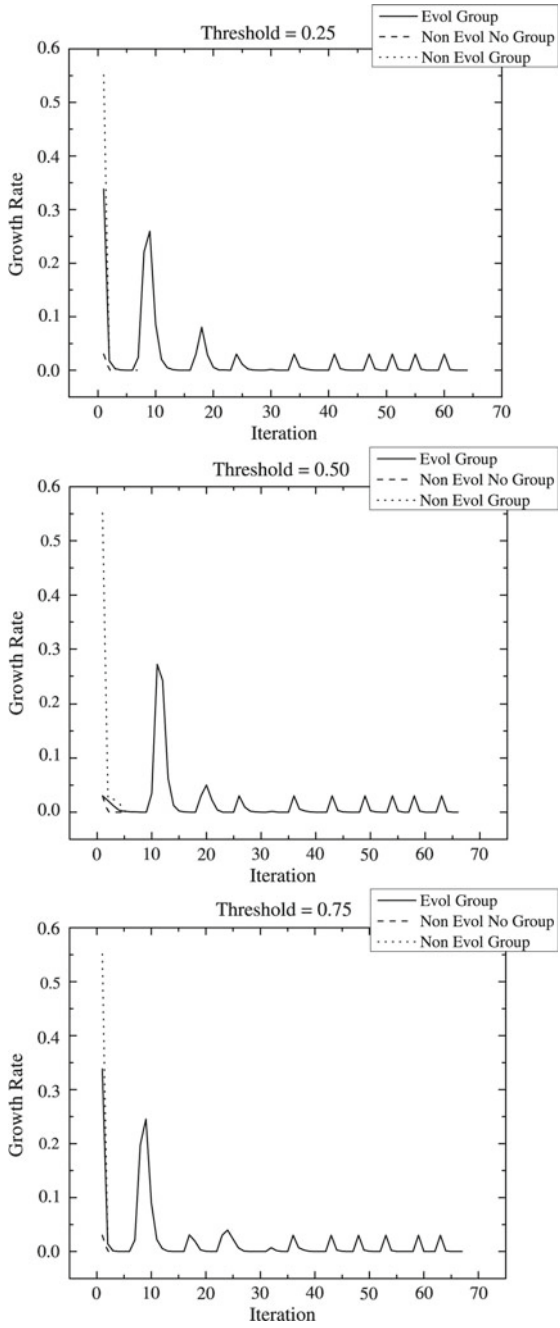

Fig. 18 Growth rate versus threshold θ

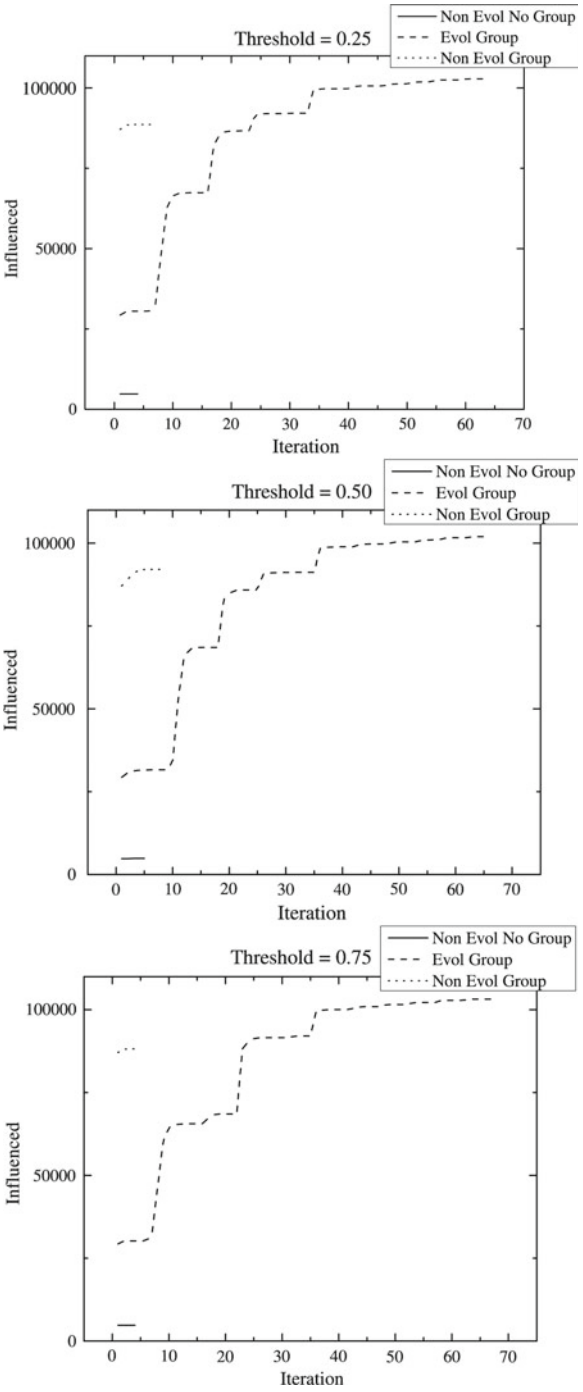

Fig. 19 # Influenced user versus threshold θ

analyzing the relationship between the users. Aalst and Song [16] proposed the concept of process mining in social network analysis to determine the relationship between the users based on their interactions. The concept makes the use of a triplet consisting of case, activity, and user. If an event (c1, a1, u1) followed by (c1, a2, u2) represents an interaction between user u1 and u2 for the case c1. However, if there is another user u3 who shares the same responsibility as u2 but there is no event of u3 on the same case following u1's activity then it depicts that u1 and u2 are closely bonded in the network as compared to u1 and u3.

A user interacts with only a small subset of the users in the entire network. It is necessary to identify the dense relationship between different subset of users in the network. These dense relationships can be represented as a group in the network. Leskovec et al. [17] analyzed that 77 % of the recommendations comes from within the group. The identification of such groups is necessary to select users from each group to facilitate the flow of influence in viral marketing. Long et al. [18] proposed a technique of identifying such groups in a network by addition of a virtual node. This node represents group features identified from the empirical data. Using Gaussian similarity, the users in the network that share those features are connected to those virtual nodes forming a group. Chi et al. [19] proposed a concept for the group identification using community factorization method. They proposed the method on social networks like blogosphere where the structural and temporal dynamics of the graph is different than the Web with short life time dens subgraph. The factorization method extracts the communities and their temporal behavior and assists in identifying the long-term graph structure from a series of short-term graphs.

Previous analyses of the social network for marketing were based on static graphs with data captured over a period of time. These analyses do not take into account the temporal changes. Social relationships change with time, with the addition or removal of users from the network and change in the relationships between the users. These changes affect the flow of influence in the network. Thus, integrating the temporal nature of the graph is of prime importance. Evolutionary clustering proposed by Chakraborti et al. [5] provided a new dimension for clustering in a dynamic graph. The gist of this concept is that the clustering should be faithful to the current data and should not deviate dramatically from the previous time step. The concept proposes the computation of sequence of clusters in each time window. The cost of the clustering is represented as the combination of the snapshot quality and the history cost. The model considers the object feature similarity along with the time-series similarity function. The model penalizes for the deviation in clustering with respect to the history data but not with respect to the new data. The input for the model is the matrix representing the relation between each pair of objects at each time step and the output is the clustering with respect to the new matrix and history. Chakraborti et al. [5] model is restricted to clustering which can be adapted for various dynamic graph analyses like user selection for viral marketing in dynamic graph. Sharan and Neville [20] worked on temporal relationships for predictive analysis in dynamic graphs. They summarized the dynamic graph with the weighted static graphs and then incorporated the weighted links in the Relational Bayes Classifier to moderate the influence of the attributes. The model tries to exploit both the relational and

temporal aspects in the domain of predictive data mining. It is based on the concept of homophily in relational domains to mine inference about nature of relationship with the recent ones conferring more homophily than earlier one.

It is necessary to efficiently predict the flow of the influence in a dynamic graph to compare the states of the network at different time steps to conform to the evolutionary concept. Kempe et al. [6] provided an approximation model to predict the flow of influence in the network. They proposed two approaches viz. linear and probabilistic. The linear approach maintains a threshold value of influence for each user. An inactive user (not influenced) gets influenced by an active user (influenced) if the summation of the weights of direct active friends goes beyond the threshold value for that user. Here the weight represents the ability of an active user to influence an inactive user. The independent cascade model follows a probabilistic approach. This model allows an active user to influence an inactive user in only one step. The probability of the inactive user getting influenced because of the active user is based on empirical data. If the inactive user does not get influenced in the following steps that active user does not get another chance to influence that inactive user.

6 Conclusion and Future Work

We considered the problem of selection of the users in a dynamic network to maximize the flow of influence and proposed a threshold-based evolutionary framework. This framework selects the users in a network by conforming to the evolutionary principle and prolonging the flow of influence by dynamically adapting to the change in the network. Experiments on HEP-Th, Friendfeed, and Epinion demonstrated the superiority of this model in comparison to the nonevolutionary models. The proposed model not only maximizes the influence flow but also dynamically adapts to the change in the network, thereby maintaining the flow in the network.

The current framework was implemented with threshold-based models. It would be interesting to study the implementation of probability-based models. The calculation of the network values was based on quantitative parameters. This can be extended to include subjective parameters. The framework can be extended to incorporate text mining on the data to provide more intuitive information for determining the relationship between the users in the network.

References

1. Domingos P (2005) Mining social networks for viral marketing. IEEE Intell Syst 20:80–82
2. Merkle (2010) View from the social inbox 2010. http://www.merkleinc.com/user-assets/Documents/WhitePapers/Social%20Inbox%202010%20WPaper%20Final.pdf
3. Gen Y (2009) Study shows Gen Y wants more control in email exchanges
4. Epsilon (2008) Asia Pacific consumer email survey. Technical report

5. Chakrabarti D, Kumar R, Tomkins A (2006) Evolutionary clustering. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, pp 554–560
6. Kempe D, Kleinberg J, Tardos V (2003) Maximizing the spread of influence through a social network. In: International conference on knowledge discovery and data mining, pp 137–146
7. Xu X, Long B, Zhang Z, Yu PS (2007) Community learning by graph approximation. In: Proceedings of the seventh IEEE international conference on data mining, pp 232–241
8. Xiang T, Gong S (2008) Spectral clustering with eigenvector selection. Pattern Recognit 41:1012–1029
9. Hopcroft J, Tarjan R (1973) Efficient algorithms for graph manipulation. Commun ACM 16:372–378
10. Michael F, David HC, Boris I (1989) Some implementations of the boxplot. Am Stat 43:50–54
11. Celli F, Di Lascio FML, Magnani M, Pacelli B, Rossi L (2010) Social network data and practices: the case of friendfeed. In: International conference on social computing, behavioral modeling and prediction, Berlin (2010)
12. Hep-th—kdl—umass amherst. https://kdl.cs.umass.edu/download/attachments/3440884/hepth-schema.png?version=1&modificationDate=1345733950033
13. Massa PAP (2006) Trust-aware bootstrapping of recommender systems. In: Proceedings of ECAI 2006 workshop on recommender systems, pp 29–33
14. Richardson M, Domingos P (2002) Mining knowledge-sharing sites for viral marketing. In: Eighth international conference on knowledge discovery and data mining, pp 61–70
15. Domingos P, Richardson M (2001) Mining the network value of customers. In: Seventh international conference on knowledge discovery and data mining, pp 57–66
16. Aalst WMvd, Song M (2004) Mining social networks: uncovering interaction patterns in business processes. In: Desel J, Pernici B, Weske M (eds) Business process management. Springer, Berlin, pp 244–260
17. Leskovec J, Adamic LA, Huberman BA (2007) The dynamics of viral marketing. ACM Trans Web (TWEB) 1:5-228–5-237
18. Long B, Xu X, Yu PS, Zhang Z (2007) Community learning by graph approximation. In: Proceedings of the 2007 seventh IEEE international conference on data mining, pp 232–241
19. Chi Y, Zhu S, Song X, Tatemura J, Tseng BL (2007) Structural and temporal analysis of the blogosphere through community factorization. In: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining, pp 163–172
20. Sharan U, Neville J (2007) Exploiting time-varying relationships in statistical relational models. In: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on web mining and social network analysis, pp 9–15

Mining and Analyzing the Italian Parliament: Party Structure and Evolution

Alessia Amelio and Clara Pizzuti

Abstract The roll calls of the Italian Parliament in the XVI legislature are studied by employing multidimensional scaling, hierarchical clustering, and network analysis. In order to detect changes in voting behavior, the roll calls have been divided in seven periods of six months each. All the methods employed pointed out an increasing fragmentation of the political parties endorsing the previous government that culminated in its downfall. By using the concept of modularity at different resolution levels, we identify the community structure of Parliament and its evolution in each of the considered time periods. The analysis performed revealed as a valuable tool in detecting trends and drifts of Parliamentarians. It showed its effectiveness at identifying political parties and at providing insights on the temporal evolution of groups and their cohesiveness, without having at disposal any knowledge about political membership of Representatives.

Keywords Social network analysis · Italian parliament · Data mining · Genetic algorithms · Political parties

1 Introduction

In the last years political parties in Italy have been affected by a steady fragmentation, with a high number of Parliamentarians leaving the group that allowed them to be elected to join another one, often changing party many times.

In this paper we investigate Italian Parliament by using different tools coming from Data Mining and Network Analysis fields with the aim of characterizing the modifications Parliament incurred, without any knowledge about the ideology or

A. Amelio (✉) · C. Pizzuti
Institute for High Performance Computing and Networking (ICAR),
National Research Council of Italy (CNR),
Via Pietro Bucci, 41C, 87036 Rende, CS, Italy
e-mail: amelio@icar.cnr.it

C. Pizzuti
e-mail: pizzuti@icar.cnr.it

Ö. Ulusoy et al. (eds.), *Recommendation and Search in Social Networks*,
Lecture Notes in Social Networks, DOI 10.1007/978-3-319-14379-8_12

political membership of its Representatives, but relying only on the votes cast by each Parliamentarian. We consider the roll calls of the period of three years and an half from April 2008 until October 2011, after which there was the fall of the center-right coalition that won the elections. This period has been equally divided in seven semesters and the votes cast by each Parliamentarian have been stored. Note that in our analysis we do not consider the Italian Senate.

Voting records have been used in two different ways. In the first approach we directly use them to show party cohesion during the considered period, and apply a multidimensional scaling technique to reveal political affinity of Parliamentarians, independently of their true party membership. This kind of analysis is interesting because it is able to reproduce the effective political alliances, without assuming parties as relevant clusters.

In the second one, from voting records we compute similarity between each pairs of Representatives and try to detect structural organization and evolution of Parliament by applying data mining and network analysis techniques. In particular, similarity among Parliamentarians is exploited to perform clustering by employing agglomerative hierarchical clustering. The division of Representatives in groups is congruous with that obtained by applying multidimensional scaling, thus showing the robustness of both approaches.

As regards network analysis techniques, from the similarity matrix a network is built where nodes correspond to Parliamentarians and an edge between two nodes exists if the similarity between the corresponding Representatives is above a fixed value. Topological features characterizing the network are studied by computing some well-known measurements to quantify structural properties, and community detection is applied to study the organization of members in groups. By using the modularity concept [9], we identify communities of members who voted similarly, and investigate how the party cohesion evolves along the semesters. The analysis provides an explicit and clear view of the steady fragmentation of the coalition endorsing the center-right government that caused the majority breakdown. Thus modularity allows a more deep analysis of the internal agreement of parties, and demonstrated a powerful means to give insights of changes in majority party.

The investigation of voting records with computational techniques is not new [6, 8, 10, 12, 19, 20], though this is the first study regarding an Italian institution.

The paper is organized as follows. In the next section we give a brief description of the Italian Parliament organization and the data set used for the analysis. In Sect. 3 we describe the voting matrix, compute party cohesion, and apply multidimensional scaling approach to voting records. In Sect. 4 the similarity metric used is defined, and the groups obtained by applying hierarchical clustering are showed is Sect. 5. Section 6 builds Parliamentarian networks, identifies and visualizes voting record blocks along the semesters. Section 7 computes measurements, well-known in network analysis, to study the characteristics of Parliamentarian network. Section 8 investigates community structure. Section 9 argues about the results obtained for the last semester. Section 10 gives a description of related work. Section 11, finally, concludes the paper and outlines future developments.

2 Data Description

The Italian Parliament of XVI legislature has been elected in April 2008 and it is constituted by 630 representatives originally elected in five main political parties: People of Liberty (PDL), League of North (LN), Democratic Party (PD), Italy of Values (IDV), and Democratic Union of Center (UDC). The majority of center-right that governed Italy until November 2011 was composed by the first two parties. To better understand the analysis we performed, it is important to know that two main events characterized the political organization of Parliament: (1) in July 2010 a group of Representatives divided from PDL to form a new political party named Future and Liberty (FL); (2) in December 2010 some Parliamentarians, mainly coming from the center-left coalition, separated from their party to constitute a new coalition, named People and Territory (PT), that endorsed the center-right government, allowing it to rule the country for other almost 10 months. Furthermore, along all the three years and a half, several Representatives abandoned their party to move in a group called Mixed. The Italian Parliament maintains a database of the legislative activity by storing, for each bill voted, the list of votes cast by each Representative. From the web site http://parlamento.openpolis.it it is possible to download the voting record of each Parliamentarian, together with some personal information, such as territorial origin, and actual group membership. For every roll call, the Openpolis database stores the vote of each Parliamentarian in three ways: "yes," "no," and "not voting." This last kind of vote can be due to either absence or abstention, but they are treated in the same manner.

3 Analysis of Voting Patterns

We collected the roll calls of the Italian Parliament in the period starting from April 2008 until October 2011, after which there was the fall of the center-right coalition that won the elections. This period of three years and a half has been equally divided in seven semesters and the votes cast by each Parliamentarian have been stored in matrices of size $n \times m$, where n is the number of Parliamentarians, and m is the number of bills voted in the reference period. Since some Parliamentarians, for several reasons, never voted, they have been eliminated. Thus the number n of Representatives reduced to 612. As regards m, it assumes a different value, depending on the semester. The number of bills voted is reported in Table 1. Seven voting matrices have been built in the following way: an element A_{ij} of a voting matrix A is $+1$ if

Table 1 Number of voted measures for each semester

I	II	III	IV	V	VI	VII
386	422	328	343	373	332	89

the Representative i voted "yes" on measure j, -1 if he voted "no," and 0 if he did not vote. The voting matrices are exploited to study the voting behavior of the Italian Parliament in two different ways. In the first approach we use them to compute party cohesion and to characterize the political affinity of Parliamentarians, independently of their true party membership. In the second one, we compute similarity for each pairs of Representatives and try to detect structural organization and evolution by applying hierarchical clustering and community detection based on the concept of modularity.

3.1 Party Cohesion

Given the voting matrices, the first investigation that can be done is to compute the cohesion of each political party along the considered period and compare the results obtained. To this end, the *agreement index* [4] measures the level of cohesion within a party by exploiting the number of equal votes for each roll call. The agreement index for each roll call is defined as follows:

$$\mathrm{AI}_i = \frac{\max\{y_i, n_i, a_i\} - \frac{y_i + n_i + a_i - \max\{y_i, n_i, a_i\}}{2}}{y_i + n_i + a_i} \tag{1}$$

where y_i is the number of members who voted "yes" in the voting i, n_i is the number of members who voted "no," and a_i is the number of members who did not vote. Group cohesion is then computed as the average of agreement indices for all the roll calls:

$$\mathrm{AI} = \frac{\sum_i^m \mathrm{AI}_i}{m} \tag{2}$$

The agreement index ranges from 0 (complete disagreement) to 1 (complete agreement).

Figure 1 displays the trend of agreement index of the five main political parties during the seven semesters. It is clear from the figure that the opposition parties show an increasing cohesion, while PDL, that started with a value near to 0.9, has a constant downtrend until the sixth semester, with a slight increment in the last semester. The variation of internal cohesion well reflects the actual political situation along the considered periods.

3.2 Singular Value Decomposition

We now analyze the voting behavior of Italian Parliament by applying the well-known multidimensional scaling technique Singular Value Decomposition (SVD) [15], whose advantages with respect to other techniques have been discussed in [1].

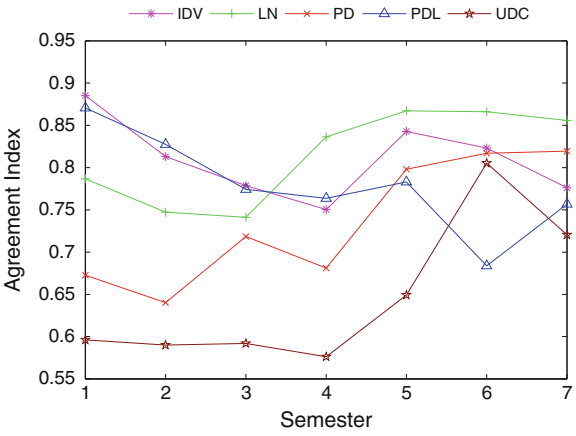

Fig. 1 Agreement index of parties for all the semesters

Let A be an $n \times m$ voting matrix where rows correspond to Representatives and columns to the votes cast to approve a law. The *Singular Value Decomposition* of A is any factorization of the form

$$A = U \times \Lambda \times V^T \tag{3}$$

where U is an $n \times n$ orthogonal matrix, V is an $m \times m$ orthogonal matrix, and Λ is an $n \times m$ diagonal matrix with $\lambda_{ij} = 0$ if $i \neq j$. The diagonal elements λ_i are called the *singular values* of A. It has been shown that there exist matrices U and V such that the diagonal elements of Λ are the square roots of the nonzero eigenvalues of either AA^T or A^TA, and they can be sorted such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ [15]. Geometrically, this factorization defines a rotation of the axis of the vector space defined by A where V gives the directions, Λ the strengths of the dimensions, and $U \times \Lambda$ the position of the points along the new axis. Intuitively, the U matrix can be viewed as a similarity matrix among the rows of A, i.e., the Representatives, the V matrix as a similarity matrix among the columns of A, i.e., the votes cast for each law, the Λ matrix gives a measure of how much the data distribution is kept in the new space [5]. If the singular values λ_i present a fast decay, then $U \times \Lambda$ provides a good approximation of the original voting matrix A. In particular, by projecting on the first two coordinates, we obtain a compressed representation of the voting matrix that approximates it at the best. The visualization of the projected approximation matrix, allows to identify groups of Representatives that voted in a similar way on many bills. As observed in [12], the first coordinate correlates to party membership, thus it is called the *partisan* coordinate. The second coordinate correlates to how often a Representative voted with the majority, thus it is called the *bipartisan* coordinate.

Figure 2 shows the application of SVD on the voting records of the Italian Parliament for the first six semesters of the current legislature. Each point corresponds to the projection of votes cast by a single Parliamentarian onto the leading two eigenvectors partisan and bipartisan. Each party has been assigned a different color and

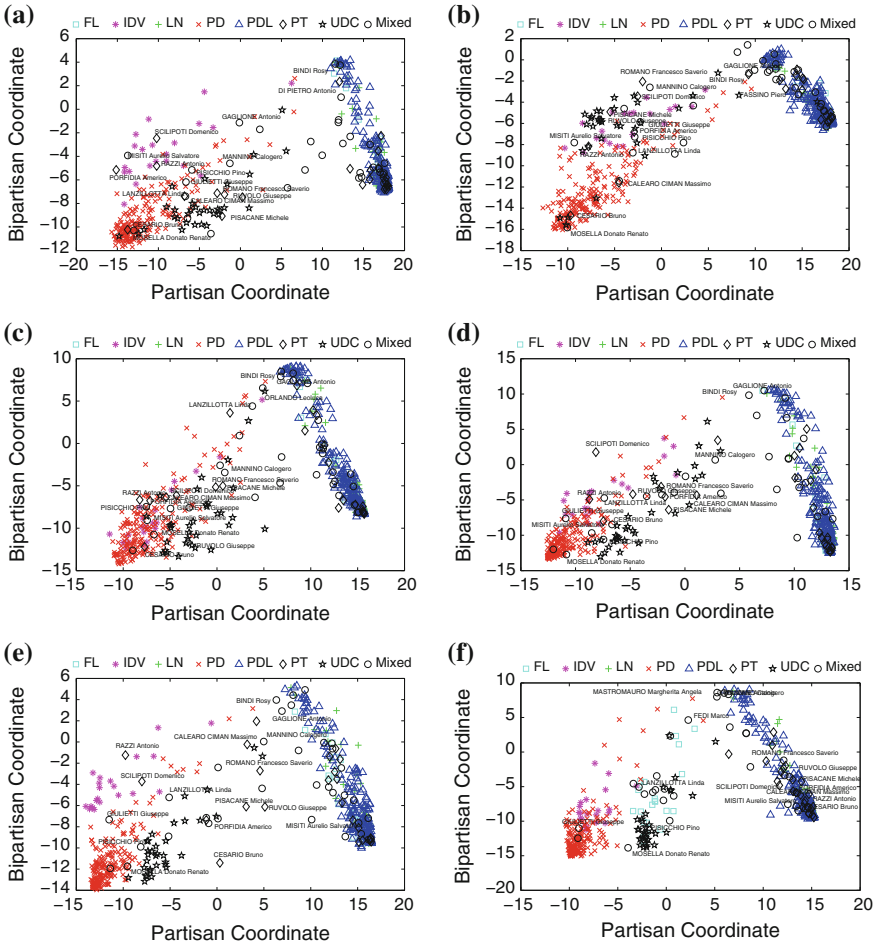

Fig. 2 Singular value decomposition of the Italian Parliament voting behavior for each of the six semesters starting from April 2008 until March 2011. **a** I Semester: April-September 2008. **b** II Semester: October 2008-March 2009. **c** III Semester: April-September 2009. **d** IV Semester: October 2009-March 2010. **e** V Semester: April-September 2010. **f** VI Semester: October 2010-March 2011

symbol. The main objective of this analysis was to study the changes in voting behavior of those Parliamentarians that moved from the opposition coalition to the majority one. Thus we selected some members of PT party and Mixed group, and visualized their names on all the figures. First of all we point out that the representation of the two coalitions center-right and center-left, and their evolution along the three years, summarized by the six figures, is very impressive.

Figure 2a clearly shows a compact center-right aggregation, a less cohesive, but clearly distinguishable, center-left alliance, and a strong connected PD subgroup

(left bottom). It is worth to note that this subgroup maintains its connectedness for all the time periods, with a slight dispersion in the second semester. The same cohesiveness is shown by PDL and LN, as expected. Moreover FL, which was included in PDL until July 2010, demonstrated its political disagreement in the sixth semester by coming nearer to UDC, as effectively happened. As regards the chosen members of PT and Mixed group, we can observe a steady movement from the center-left coalition to the center-right one since the fourth semester. This shift is much more evident in the fifth semester, when the voting behavior of these Representatives approached closer and closer to center-right majority. In fact, all the Parliamentarians located in the central part of Fig. 2e, appear at right in Fig. 2f, indistinguishable from the majority coalition.

We also notice that there is a PD Parliamentarian positioned upper, near the right coalition, for five semesters. Because of the interpretation of the bipartisan coordinate, her location means that she mostly voted with the majority. This dissimilarity from the own political party, perhaps can be explained by the fact that this Representative was vice president of the chamber.

Analysis of voting behavior with Singular Value Decomposition is thus a powerful tool to characterize political ideology of Parliamentarians, and to trace the evolution of their position along consecutive time periods. SVD is able to find structural patterns and latent information in the voting records without any knowledge about the political orientation of Representatives.

4 Parliamentarians Similarity

There can be different ways of defining similarity between two Parliamentarians from the voting matrix. For example, Jakulin and Buntine [6] used the mutual information concept. However, as observed by the authors, if two members always vote in the opposite way, they also are considered similar. We think that this kind of proximity measure misrepresents the Representative closeness, thus we employed a more suitable measure. Considering that when two Representatives cast a vote, the values "yes" and "no" should be considered equally important in comparing their political affinity, we adopted the proximity measure known as *simple matching coefficient* (SMC) [16]. We ignored the cases when at least one of the two did not vote because, as already pointed out, this means either abstention or absence, and we cannot distinguish between them. Thus there can be four different outcomes: (1) yy, both voted "yes," (2) nn, both voted "no," (3) yn, the first Parliamentarian voted "yes" and the second one "no", (4) ny, the first Parliamentarian voted "no" and the second one "yes." Then the SMC of Parliamentarians p_1 and p_2 is defined as

$$\mathrm{SMC}(p_1, p_2) = \frac{yy + nn}{yy + nn + yn + ny} \tag{4}$$

The simple matching coefficient thus computes the fraction of equal votes, both "yes" or "no," with respect to the total votes they cast. The similarity metric defined allows us to measure the closeness of each pair of Parliamentarians on the base of their voting behavior. In such a way a symmetric similarity matrix M among all the Parliamentarians can be built, and their proximity with the members of the same or opposite parties studied. A summarized view of the affinity between each couple of Representatives can be done in different ways. In the following we first apply a hierarchical clustering algorithm, and then we give a graphical representation of the similarity matrix.

5 Hierarchical Clustering

We apply the agglomerative hierarchical clustering method known as *single linkage clustering* [16]. The algorithm uses the smallest distance between two Parliamentarians and it generates a hierarchical cluster tree known as *dendrogram*. The dendrogram shows the cluster–subcluster relationships and the hierarchical structure of the merged groups. Figure 3 represents very well the political alliances along all the semesters.[1] The colors inside the dendrogram represent the clusters found by the algorithm. Attached to the leaves there are the names of the corresponding politicians, painted with the colors of the true associated party.

In Fig. 3a we can observe as the two main political parties, PD in red and PDL in blue, correspond to the two main clusters of the dendrogram for all the semesters. The other parties (IDV in magenta, FL in cyan, LN in green, PT in orange, UDC in brown, and Mixed in violet) are clusters of smaller size, or they are merged inside the main clusters. For example, LN party is grouped together with PDL in all the semesters, reflecting the real political (center-right) alliance between PDL and LN. Another similar case is IDV: most of the members are grouped with the PD, while some of them appear in different clusters for all the semesters.

Let us now consider the remaining parties. FL, as already described, was included into PDL until July 2010, when internal problems caused the movement of FL in the direction of center-left alliance. This phenomenon is captured from the clustering process. In fact FL is included into the majority for the semesters I–V (Fig. 3a–e), while in the sixth semester all the members of FL are separated from PDL and grouped together with the opposite part (Fig. 3f).

In order to analyze more clearly the trend of PT and Mixed parties, we looked not only at the dendrograms but also at the confusion matrices generated for all the semesters. They show what really happened along the semesters of the legislature: the gradual movement of PT and of some members of the Mixed group in the direction of the center-right alliance.

[1] Enlarged figures of all the dendrograms can be downloaded from https://sites.google.com/site/alessiaamelio/software-tools.

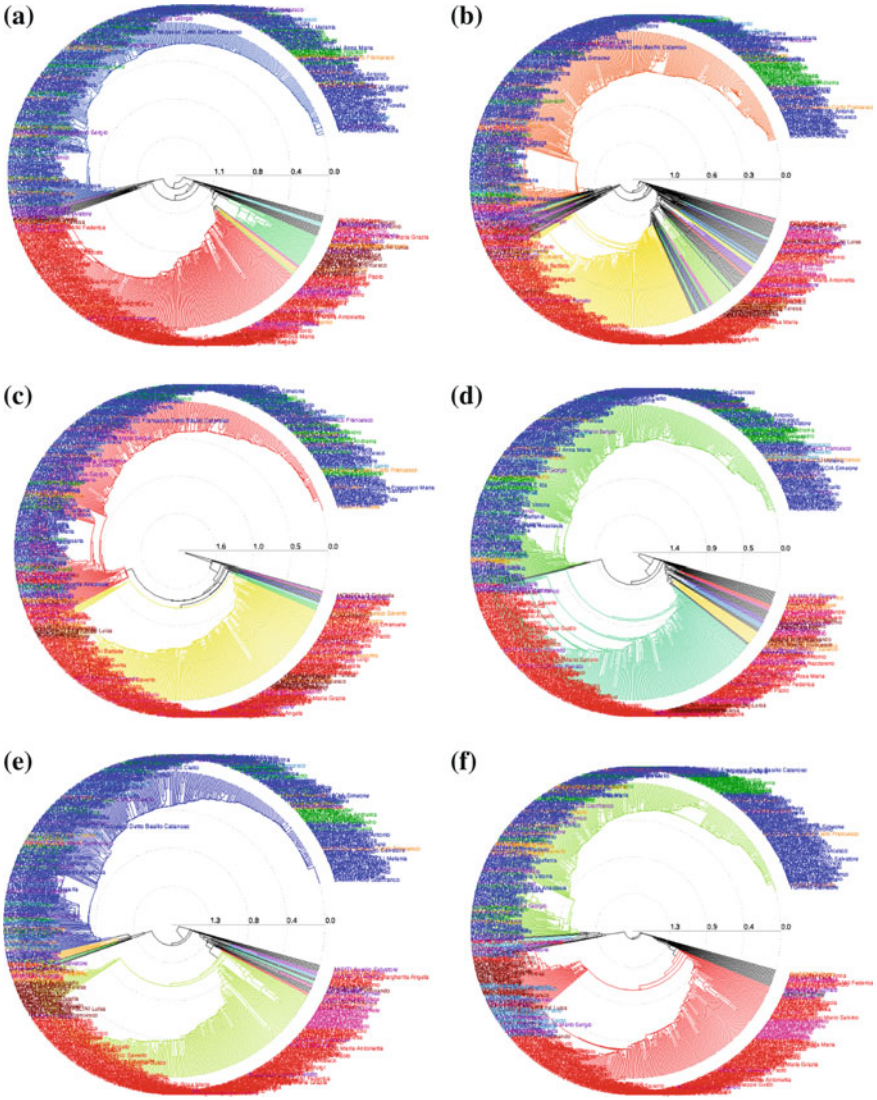

Fig. 3 Dendrograms obtained by the single linkage clustering algorithm for each semester. Internal colors correspond to the clusters found by the algorithm, external colors to the true parties. The association color party is the following: FL: *cyan*, IDV: *magenta*, LN: *green*, PD: *red*, PDL: *blue*, PT: *orange*, UDC: *brown*, Mixed: *violet*. **a** I Semester. **b** II Semester. **c** III Semester. **d** IV Semester. **e** V Semester. **f** VI Semester

Furthermore, it is interesting to observe that UDC is recognized from the clustering process as a group (Fig. 3a), while in the sixth semester (Fig. 3f) it appears together with FL and grouped with PD. This is due to the political alliance between the UDC and FL and to the movement of both parties in the direction of the center-left alliance.

It is worth to note as the main voting patterns revealed by hierarchical clustering totally agree with the results of the SVD analysis performed in the previous section.

6 Network Representation of Italian Parliament Voting Records

Given the similarity matrix M among Parliamentarians, a network $\mathcal{N}$ can be built from M, by considering each Parliamentarian as a node of a weighted undirected graph $G = (V, E, \omega)$, where V is the set of vertices, E the set of edges, and $\omega : E \mapsto \Re$ is a mapping that assigns a weight to the edge (i, j), between the vertices i and j. The weight $\omega(i, j)$ corresponds to the similarity value M_{ij} between Parliamentarians i and j. M can thus be considered the weighted adjacency matrix of G. Since weights of value zero are rather rare, the graph G is an "almost" complete graph, and the application of network analysis methods could not uncover interesting properties. In order to study and investigate the Parliamentarian network, a thresholding operation on M has been considered, i.e., fixed a threshold δ, let B_δ be the adjacency matrix of G obtained by assigning a value equal to 1 to a generic element B_{ij} of B if the corresponding value $M_{ij} \geq \delta$, 0 otherwise. In the following, the subscript δ is omitted from the binary matrix B, when the value used for δ is clear from the context.

6.1 Block Visualization

In order to visualize the similarity matrix M, we considered the binarized matrix B with $\delta = 0.6$. B has been then reordered such that Parliamentarians of the same party are located as consecutive rows/columns.

Figure 4 shows how the two political parties PDL (rows 304:521) and LN (rows 45:98), that supported the center-right government, progressively reduce their intragroup similarity, while the opposition parties PD (rows 99:303), IDV (rows 24:44), and UDC (rows 546:578) present the opposite trend, i.e., in the first three semesters their intragroup similarity slightly diminishes, in the second three semesters, on the contrary, it increases. It is interesting to note that members of FL (rows 1:23) maintain their high similarity for all the periods, although they separated from PDL in 2010. Another important observation regards the new formed group PT, whose Representatives come from the center-left parties. Although this was constituted in the sixth semester to avoid the government fall, its members showed a good political affinity since the first semester (rows/columns 522:545). The figures clearly show the boosting of agreement from the first to the last semester.

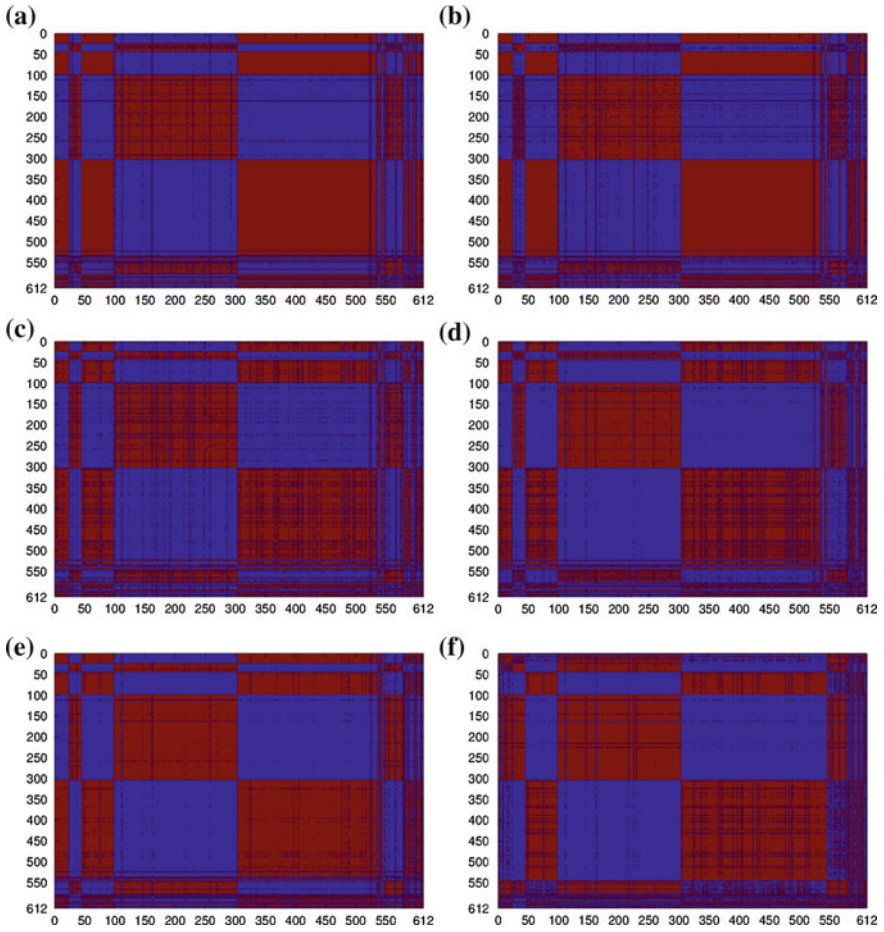

Fig. 4 Visualization of the binary similarity matrices sorted by party membership for each of the six semesters. The row/column intervals corresponding to each party are the following: FL [1:23]; IDV [24:44]; LN [45:98]; PD [99:303]; PDL [304:521]; PT [522:545]; UDC [546:578]; Mixed [579:612]. **a** I Semester. **b** II Semester. **c** III Semester. **d** IV Semester. **e** V Semester. **f** VI Semester

7 Analysis of Network Structure

The representation of similarity among Parliamentarians as a network $\mathcal{N}$ allows the analysis of topological features that characterize the network structure. In the following some measurements, coming from graph theory, that provide a quantitative characterization of the structural properties of the Parliamentarian networks, are reported and discussed.

As pointed out by Wasserman and Faust in [17], a main goal in a network is the detection of the *most important* or *central* nodes. Measures introduced by researchers

to interpret the concept of *centrality* are based on the position of nodes in the network. Important nodes are usually located in strategic positions within the network. *Degree* and *betweenness* are two concepts, explained below, that try to quantify the importance of nodes. We first consider indices based on the degree concept.

7.1 Density, Degree Centrality, and Average Degree

A node that has many ties has a central role since it can quickly exchange information with the other nodes of the network. The simplest index of centrality is the number of neighbors of a node, i.e., its degree. The degree k_i of a vertex i is defined as:

$$k_i = \sum_j B_{ij} = \sum_j B_{ji} \tag{5}$$

that is the number of edges connected to i.

Other two measures of connectedness are *average degree* and *density*. The *average degree* $\langle k \rangle$ is defined as:

$$\langle k \rangle = \frac{1}{\mid V \mid} \sum_i k_i = \frac{1}{\mid V \mid} \sum_{ij} B_{ij} \tag{6}$$

i.e., it is the average of the degrees for all vertices in the network.

The *density* d of a network is the number of links in the graph, expressed as a proportion of the maximum possible number of links.

$$d = \frac{2 \mid E \mid}{\mid V \mid (\mid V \mid -1)} \tag{7}$$

Network density depends on the size of the network. A more useful measure that allows to evaluate the structural cohesion of a network, independently of its size is the *average degree*.

These last two concepts are strictly related. If a network has low density, the average connectivity of its vertices, i.e., $\langle k \rangle$, is low, thus there can be many isolated nodes and small connected components. As the number of edges increases, the connectivity increases too, until a unique component in which the vertices are connected to each other is present.

This behavior can be observed in Fig. 5, where density and average degree are computed along the seven semesters for different values of the threshold δ. The figure shows that lower values of δ imply a higher connectivity, which culminates in the seven semester for $\delta = 0.5$, with an average degree above 500, and a density near 0.9. Considering that the number of nodes is 612, this means that Parliamentarians voted in a similar way in at least 50 % out of all the roll calls.

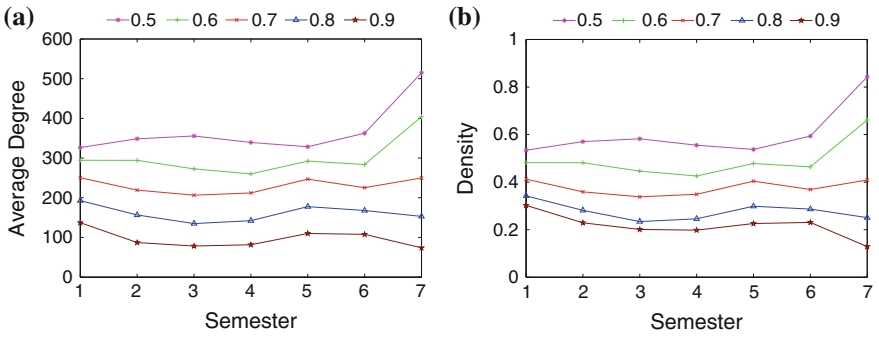

Fig. 5 Average degree (**a**) and density (**b**) on the Parliamentarian networks. Each measure is evaluated along the seven semesters by thresholding the corresponding similarity matrix M for each semester at 0.5, 0.6, 0.7, 0.8, 0.9

Degree centrality in the Parliamentarian network means that a Representative agreed with many others in voting bills, thus he/she shares political affinity with the neighbors and could influence their future voting. In Table 2 the top 20 Parliamentarians having the highest degree centrality for at least two out of the seven semesters are reported. The index has been computed by fixing the threshold $\delta = 0.6$. It is interesting to note that Claudio Scaiola (who had the role of Minister in the government) is one of the most central person, for four out of seven semesters, in particular in the last semester, while the ex-Prime Minister Silvio Berlusconi was a central node only in the first two semesters.

The set of degrees of all the nodes of a network can be summarized by a unique index named *degree centralization* that computes the variation in the degrees of vertices divided by the maximum degree variation which is possible in a network of the same size.

$$Cd(\mathcal{N}) = \frac{1}{(\mid V \mid -1)(\mid V \mid -2)} \sum_i (Cd_{\max} - Cd_i) \tag{8}$$

where $Cd_{\max}$ is the largest value of degree centrality in the network and Cd_i is the degree centrality of vertex i, corresponding to the degree k_i. Degree centralization is a measure of the dispersion of node degrees since it compares the degree of each node with the maximum degree present in the network. Its value ranges from 0, meaning that all degrees are equal, thus the graph has no variation, to 1, when a single node interacts with all the other $\mid V \mid -1$ nodes, while the other nodes are connected to only this one, which is the case of a star graph. Figure 6a shows degree centralization along the seven semesters for increasing values of threshold δ. We can note that the highest values are obtained with $\delta = 0.5$, though in the seventh semester it drastically drops, analogously to the values computed for the other thresholds. However, most of the degree centralization values range between 0.1 and 0.26, except for $\delta = 0.6$ in the sixth semester, and $\delta = 0.7$ in the seventh semester, indicating that the network is rather regular, i.e., the degrees of all nodes are similar.

Table 2 Parliamentarians with the highest degree centrality along the semesters, who are present in at least two out of seven semesters of the XVI Legislature

Name	Political party	Sem1	Sem2	Sem3	Sem4	Sem5	Sem6	Sem7
Siegfried Brugger	Mixed group	362	–	384	–	–	466	–
Stefano Stefani	LN	351	–	359	341	345	–	–
Maria Grazia Siliquini	PT	350	–	359	340	–	–	–
Adolfo Urso	Mixed group	349	371	–	337	–	–	–
Claudio Scajola	PDL	349	370	–	–	347	–	528
Carmelo Lo Monte	Mixed group	–	–	–	–	345	–	528
Karl Zeller	Mixed group	363	–	369	–	–	–	–
Silvio Berlusconi	PDL	353	372	–	–	–	–	–
Stefania Craxi	PDL	351	–	364	–	–	–	–
Michela Brambilla	PDL	350	371	–	–	–	–	–
Gianfranco Micciché	Mixed group	349	373	–	–	–	–	–
Giulio Tremonti	PDL	348	372	–	–	–	–	–
Giacomo Stucchi	LN	348	–	–	–	348	–	–
Andrea Ronchi	Mixed group	348	370	–	–	–	–	–
Stefania Prestigiacomo	PDL	348	370	–	–	–	–	–
Guido Crosetto	PDL	–	367	–	–	347	–	–
Francesco Bosi	UDC	–	–	398	–	–	400	–
Gabriella Mondello	UDC	–	–	383	–	–	385	–
Riccardo Migliori	PDL	–	–	362	–	345	–	–
Edmondo Cirielli	PDL	–	–	359	–	350	–	–
Luca Barbareschi	Mixed group	–	–	359	338	–	–	–
Giorgio Jannone	PDL	–	–	358	342	–	–	–
Giulia Cosenza	PDL	–	–	358	–	346	–	–
Francesco Pionati	PT	–	–	–	344	–	–	528

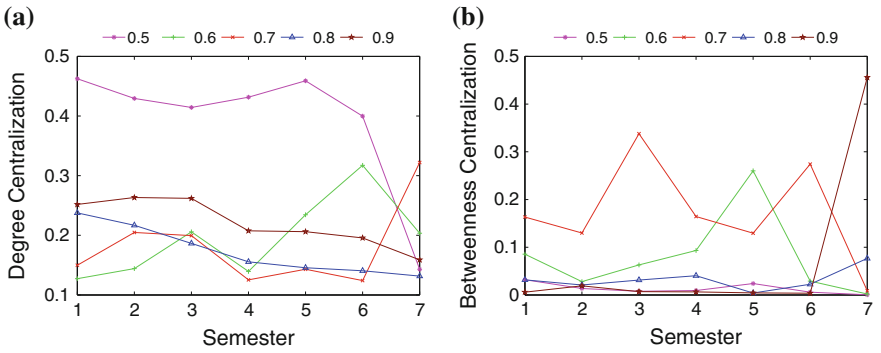

Fig. 6 Degree centralization (**a**) and betweenness centralization (**b**) on the Parliamentarian networks. Each measure is evaluated along the seven semesters by thresholding the corresponding similarity matrix M for each semester at 0.5, 0.6, 0.7, 0.8, 0.9

7.2 Betweenness

If two nodes are not directly connected with an edge, their possibility of interacting depends on the paths between them, thus on the nodes constituting these paths. A node, then, can be considered central if it appears in the shortest paths joining many of the other nodes. The *betweenness centrality* is defined as:

$$B_i = \sum_{j,k,j\neq k} \frac{n_{jk}(i)}{n_{jk}} \tag{9}$$

where $n_{jk}(i)$ is the number of shortest paths between vertices j and k that pass through vertex i, and n_{jk} is the total number of shortest paths between j and k. The *betweenness centrality* of a node measures the importance of a vertex in the network in terms of number of shortest paths in which that vertex participates.

Analogously to degree centralization, *betweenness centralization* measures the betweenness centrality variation with respect to the maximum possible variation in node betweenness, and it is defined as:

$$B_{\mathcal{N}} = \frac{1}{\mid V \mid -1} \sum_i (B'_{\max} - B'_i) \tag{10}$$

where B'_i is B_i normalized with respect to the maximum reachable value, and $B'_{\max}$ is the largest value of betweenness centrality in the network, standardized as B_i. The index reaches its maximum value, equals to 1, if the network is a star graph. In fact, in the star graph, the vertex in the middle has the highest betweenness centrality because it is on every geodesic, while all the other vertices have betweenness centrality of 0 as they are on no geodesics. On the other hand, the minimum value of $B_{\mathcal{N}}$, which is 0, occurs when all the vertices have the same betweenness centrality. Figure 6b reports the betweenness network centrality along the seven semesters for increasing values of δ. The figure points out that betweenness values are rather low except at the fifth semester for $\delta = 0.6$, the third and sixth semesters for $\delta = 0.7$, and seventh semester for $\delta = 0.9$. Tables 3, 4, and 5 report the Parliamentarians having the top 20 highest values of betweenness for the third and sixth semesters with $\delta = 0.7$, and seventh semester with $\delta = 0.9$. These people should be the most influential Parliamentarians in the network and their removal could reduce communication among the groups.

7.3 Clustering Coefficient

The clustering coefficient, also known as transitivity, expresses the idea that two friends with a common friend are likely to be friends. This concept has been defined by Watts and Strogatz in [18], and, in terms of network topology, it measures the number of triangles, i.e., the set of three vertices connected to each other. Given a

Table 3 The 20 Parliamentarians with the highest betweenness centrality in the network of the third semester with $\delta = 0.7$

Name	Political party	Betweenness centrality
Gabriella Mondello	UDC	0.3394
Francesco Laratta	PD	0.0322
Erminio Angelo Quartani	PD	0.0286
Alessandro Naccarato	PD	0.0269
Dario Franceschini	PD	0.0261
Armando Dionisi	UDC	0.0255
Gian Luca Galletti	UDC	0.0242
Angelo Compagnon	UDC	0.0240
Antonello Giacomelli	PD	0.0233
Nedo Lorenzo Poli	UDC	0.0229
Lorenzo Ria	UDC	0.0219
Michele Pompeo Meta	PD	0.0195
Roberto Rao	UDC	0.0189
Roberto Occhiuto	UDC	0.0177
Giuseppe Ruvolo	PT	0.0165
Antonio De Poli	UDC	0.0158
Nunzio Francesco Testa	UDC	0.0152
Mario Tassone	UDC	0.0144
Italo Tanoni	Mixed group	0.0142
Anna Teresa Formisano	UDC	0.0129

node i, let nt_i be the number of links connecting the k_i neighbors of i to each other. The clustering coefficient of a node i is defined as:

$$\mathrm{CC}_i = \frac{2nt_i}{k_i(k_i - 1)} \tag{11}$$

nt_i represents the number of triangles passing through i, and $k_i(k_i - 1)/2$ the number of possible triangles that could pass through node i. The clustering coefficient CC of a graph is the average of the clustering coefficients of the nodes it contains:

$$\mathrm{CC} = \frac{1}{\mid V \mid} \sum_i \mathrm{CC}_i \tag{12}$$

Clustering coefficient varies between 0 and 1. Figure 7 points out that the clustering coefficient is rather high, independently of the threshold δ used, showing thus that there are many triples of Parliamentarians voting in a similar manner. However, it is worth to note that there is no monotonicity between increasing the threshold δ and clustering coefficient values. The explanation of this behavior comes from the

Table 4 The 20 Parliamentarians with the highest betweenness centrality in the network of the sixth semester with $\delta = 0.7$

Name	Political party	Betweenness centrality
Carmine Santo Patarino	FL	0.2762
Roberto Rosso	PDL	0.2758
Luca Volonté	UDC	0.1153
Mario Baccini	PDL	0.1060
Francesco Divella	FL	0.0243
Marco Fedi	PD	0.0182
Maurizio Migliavacca	PD	0.0180
Federica Mogherini Rebesani	PD	0.0172
Angelo Compagnon	UDC	0.0117
Gabriella Mondello	UDC	0.0114
Anna Teresa Formisano	UDC	0.0114
Gianfranco Paglia	FL	0.0112
Angela Napoli	FL	0.0112
Adolfo Urso	Mixed group	0.0109
Benedetto Della Vedova	FL	0.0108
Enzo Carra	UDC	0.0108
Lorenzo Ria	UDC	0.0105
Roberto Rao	UDC	0.0102
Angelo Cera	UDC	0.0101
Pierluigi Mantini	UDC	0.0099

decreasing degree values of vertices when δ augments. In fact, when $\delta = 0.5$, average node degree is between 330 and 520, thus the number of possible triangles that could pass through a vertex is very high. This could drop the value of CC_i if the neighbors of node i are not well connected. On the other hand, for $\delta = 0.9$ both average degree and number of connections are rather low, thus clustering coefficient assumes smaller values.

7.4 p-cliques

Another important feature to study in networks is the presence of *cliques*, i.e., maximal complete subgraphs of at least three nodes. However, the request that each node must be connected with all the other nodes of the subgraph is rather strong. Thus we fix the attention on the *p-cliques*, i.e., groups of nodes having at least a proportion p of neighbors inside the same group. We considered PDL party, and computed the *p-cliques*, with $p = 0.5$. Very interestingly, we obtained a group of

Table 5 The 20 Parliamentarians with the highest betweenness centrality in the network of the seventh semester with $\delta = 0.9$

Name	Political party	Betweenness centrality
Fiamma Nirenstein	PDL	0.4639
Gaetano Porcino	IDV	0.4587
Matteo Mecacci	PD	0.4583
Massimo Parisi	PDL	0.4568
Nicodemo Nazzareno Oliverio	PD	0.4021
Arturo Iannaccone	PT	0.1973
Vincenzo Barba	PDL	0.1915
Luigi Vitali	PDL	0.1846
Paolo Bonaiuti	PDL	0.1322
Francesco Colucci	PDL	0.0811
Michela Vittoria Brambilla	PDL	0.0510
Maurizio Bernardo	PDL	0.0496
Riccardo De Corato	PDL	0.0445
Gregorio Fontana	PDL	0.0384
Valentino Valentini	PDL	0.0364
Sestino Giacomoni	PDL	0.0364
Giampaolo Fogliardi	PD	0.0355
Giorgio Merlo	PD	0.0352
Agostino Ghiglia	PDL	0.0306
Ida D'Ippolito Vitale	PDL	0.0306

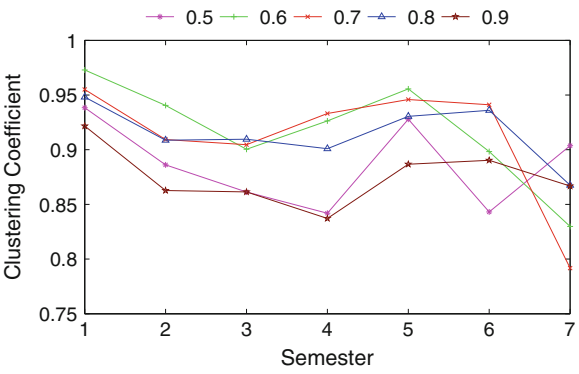

Fig. 7 Clustering coefficient of the Parliamentarian networks, evaluated along the seven semesters by thresholding the corresponding similarity matrix M for each semester at 0.5, 0.6, 0.7, 0.8, 0.9

19 Parliamentarians, reported in Table 6, that remained compact for all the seven semesters. These Representatives have main roles in the PDL party. In particular, Angelino Alfano is the party secretary, Umberto Bossi was the LN party secretary, while Roberto Maroni is the actual LN secretary and Minister of the Berlusconi's government, 9 out of 19 played the role of Minister, and the others have a main

Table 6 The 19 most faithful Parliamentarians of ex-Prime Minister

Name	Political party
ALFANO Angelino	PDL
BERLUSCONI Silvio	PDL
BONAIUTI Paolo	PDL
BOSSI Umberto	LN
BRAMBILLA Michela	PDL
CARFAGNA Maria Rosaria	PDL
CROSETTO Guido	PDL
FITTO Raffaele	PDL
FRATTINI Franco	PDL
GELMINI Mariastella	PDL
LA RUSSA Ignazio	PDL
LUPI Maurizio	PDL
MARONI Roberto	LN
MELONI Giorgia	PDL
MICCICHE' Gianfranco	PDL
PRESTIGIACOMO Stefania	PDL
ROMANI Paolo	PDL
ROTONDI Gianfranco	PDL
TREMONTI Giulio	PDL

position in PDL party. The nucleus of 19 persons can be considered the "most faithful" supporters of the ex-Prime Minister, that remained "devoted" until the end. A variable number of other Representatives joined or left this dense group of people. Figures 8, 9, and 10 show the "birth" and evolution of these p-cliques within the seven semesters. It is worth to note that in the fifth and sixth semesters many Parliamentarians disappeared from this group of faithful supporters, while in the seventh the p-clique again increased with new entries.

8 Community Structure

In this section we apply network analysis techniques to the voting records of Italian Parliament to verify if the results obtained with the approaches employed in the previous sections are comparable when changing the analysis method. To this end we consider the binary matrix B with $\delta = 0.6$. This means that two Representatives are connected if they voted in the same way in at least 60 % of the overall roll calls. The community structure of $\mathcal{N}$ can then be investigated by optimizing the well-known concept of *modularity* [9], based on the intuitive idea that a community should have

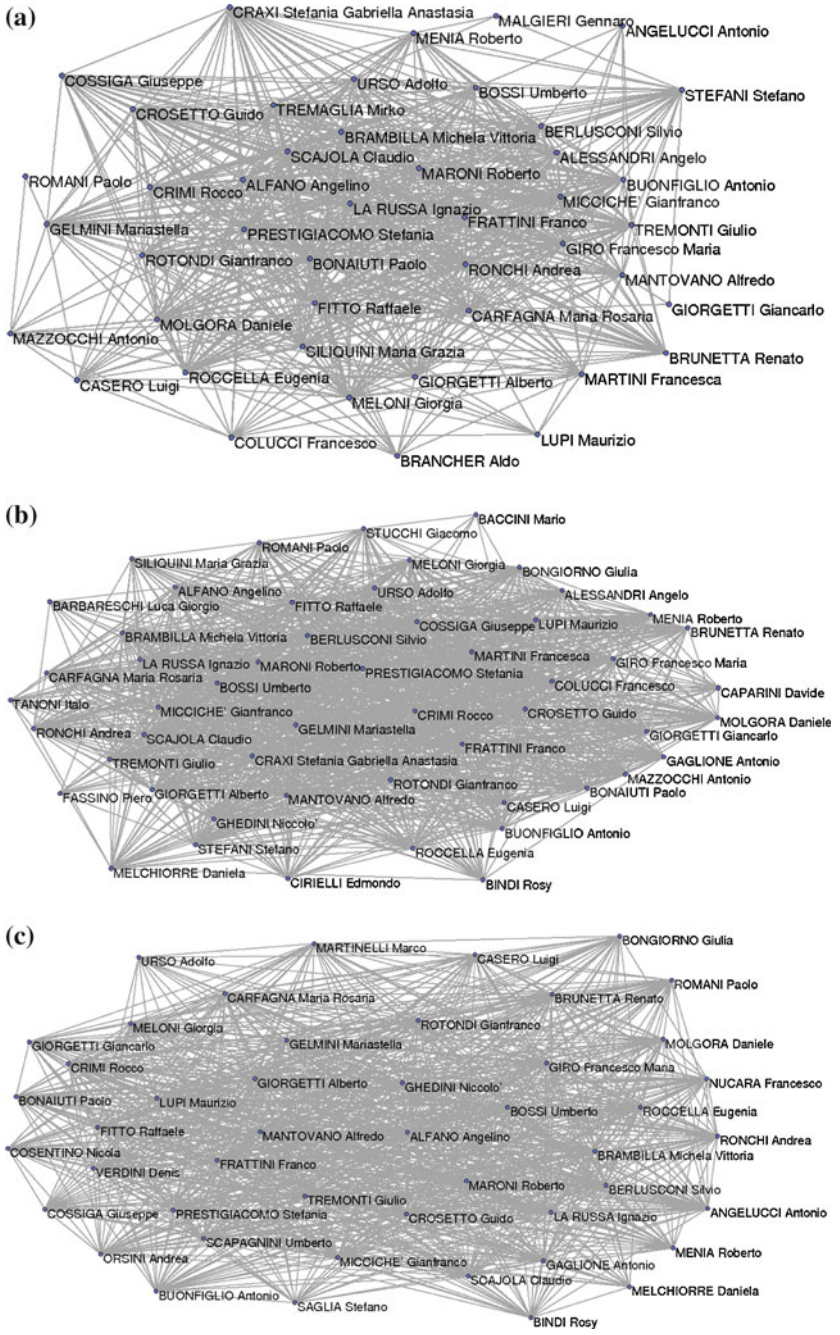

Fig. 8 p-cliques obtained for the I, II, and III semesters with number of Parliamentarians 45, 53 and 47, respectively. **a** I Semester. **b** II Semester. **c** III Semester

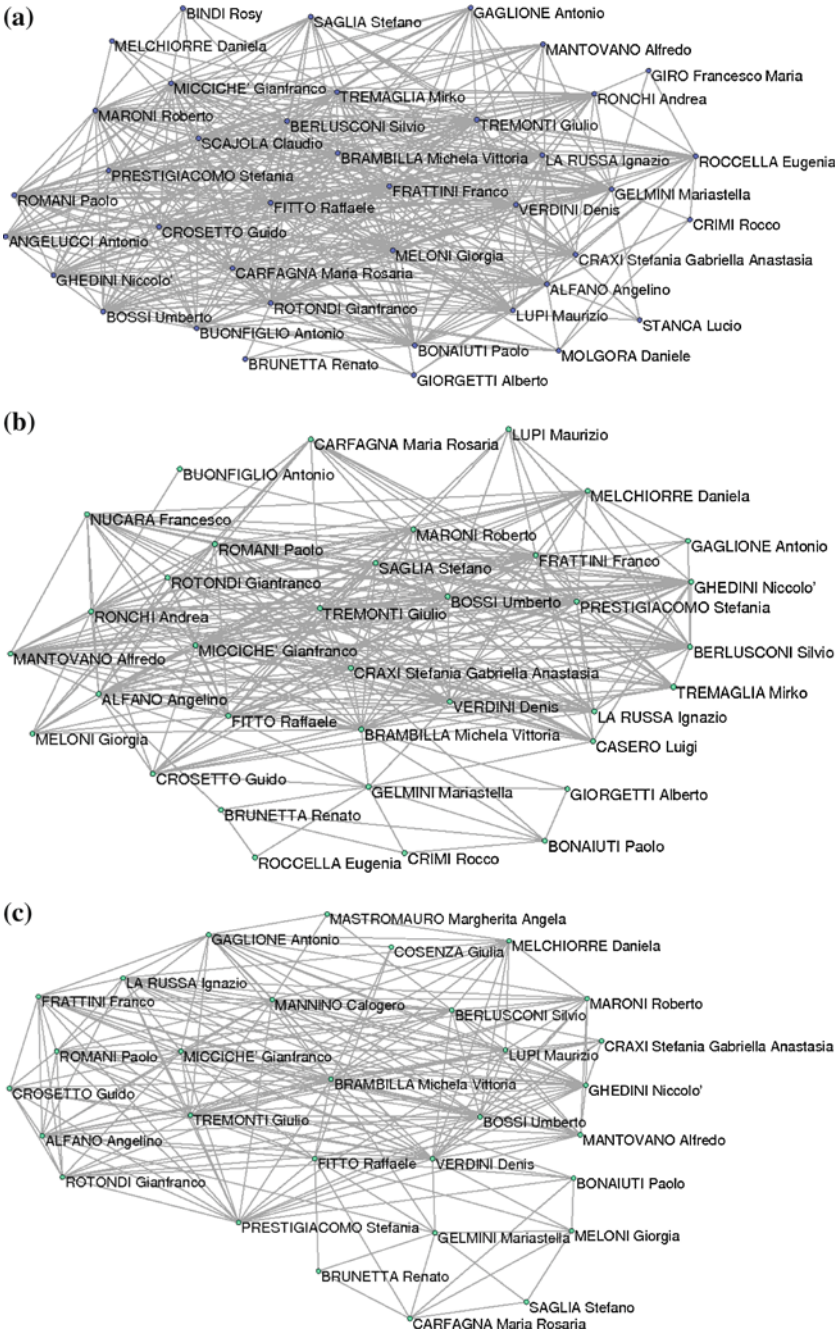

Fig. 9 p-cliques obtained for the IV, V, and VI semesters with number of Parliamentarians 39, 35 and 30, respectively. **a** IV Semester. **b** V Semester. **c** VI Semester

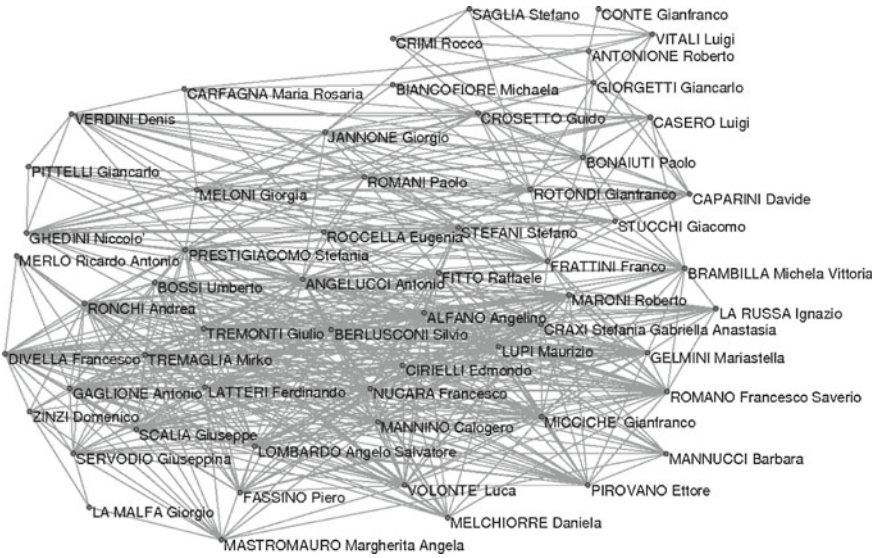

Fig. 10 p-cliques obtained for the VII semester composed by 58 Parliamentarians. **a** VII semester

more internal connections among its nodes than interconnections between its nodes and those in other communities. Modularity is defined as

$$Q = \frac{1}{2r} \sum_{ij} \left(B_{ij} - \gamma \frac{k_i k_j}{2r} \right) \delta(C_i, C_j) \tag{13}$$

where r is the number of edges in the network, k_i is the degree of node i, C_i is the community to which i belongs, and $\delta(C_i, C_j)$ is 1 if nodes i and j belong to the same community, 0 otherwise. γ is a resolution control parameter introduced by Granell et al. [3] to overcome the resolution problem stated in [2] and study community structure at multiple scales. In fact it has been proved that the optimization of modularity has a topological resolution limit that depends on both the total size of the network and the interconnections of groups. This implies that small, tightly connected clusters could not be found. Thus, searching for partitioning of maximum modularity may lead to solutions in which important structures at small scales are not discovered. When $\gamma = 1$ the equation reduces to the standard formulation of modularity [9].

We used an algorithm optimizing modularity [11] extended with the resolution parameter, and executed the method with three different values of γ: 1, 1.5, 1.9. The latter two values have been chosen to analyze the existence of subcommunities inside those obtained with $\gamma = 1$ that cannot be found by optimizing modularity because of the resolution problem.

Figure 11 shows how modularity values vary during the seven semesters for all the three resolution parameters chosen. The figure clearly points out a sharp decrease

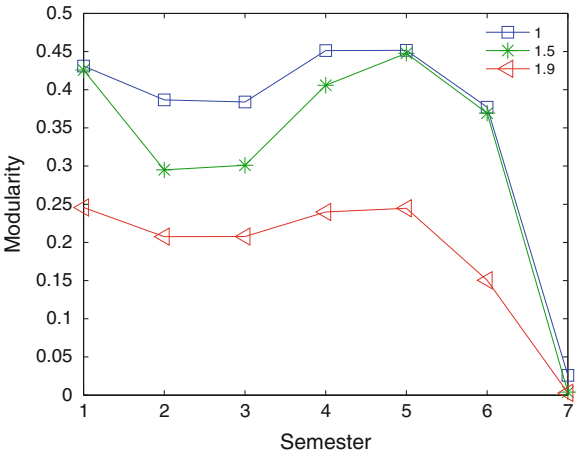

Fig. 11 Modularity for all semesters with different values of γ

of modularity in the sixth period and a drastic reduction in the seventh one. In order to better analyze the community structure detected by the algorithm, Figs. 12 and 13 show the number of communities in which the two main parties PDL and PD have been split. We do not report the results for the other parties because their behavior is analogous to the coalition they belong. Since the size of the largest community is 218 (i.e., the number of PDL members), the first coordinate varies between 1 and 218. The second coordinate, for each value of γ, reports the number of subgroups of that size obtained by the algorithm. Figure 12a shows that with $\gamma = 1$ PDL is grouped in a unique community, while PD is clustered in a big community of 190 members and other 14 members are split in seven small communities. When $\gamma = 1.5$ the situation is almost the same. However, when $\gamma = 1.9$, PD continues to have a big community of size 192, while PDL is split in 14 communities of size varying between 1 and 46. The very interesting result is that this behavior is maintained for all the semesters. Thus, while PD remains cohesive for all the semesters, independently of the γ value, PDL is divided into many subgroups since the first semester, when its degree of aggregation was considered very high, and as obtained with the other approaches described in the previous sections.

Thus modularity allows a more deep analysis of the internal agreement of parties and can provide insights of early and unexpected changes a political party could encounter. Moreover, it affords an explicit and clear view of the steady fragmentation of the coalition endorsing the center-right government that culminated in its fall.

9 The Seventh Semester

The analysis described in the previous sections mainly considered the first six semesters. We decided to separate the last semester because the voting behavior of Parliamentarians had an abrupt alteration, as testified also by the results obtained

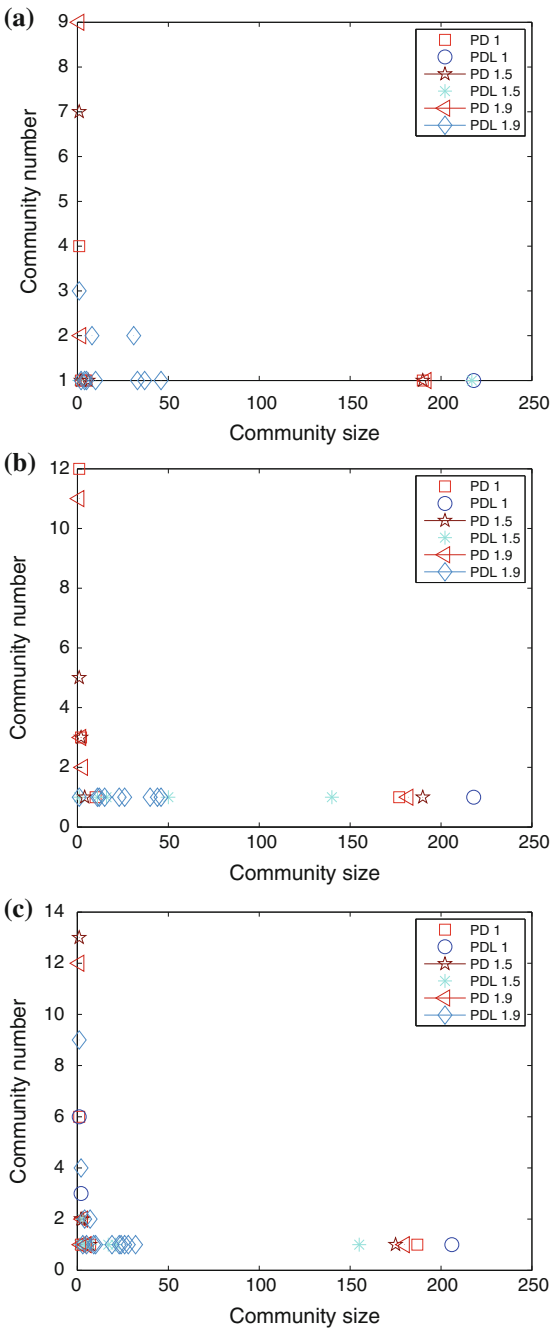

Fig. 12 Number of communities in which the two main parties PDL and PD are split and respective size for the first three semesters for different values of γ. **a** I Semester. **b** II Semester. **c** III Semester

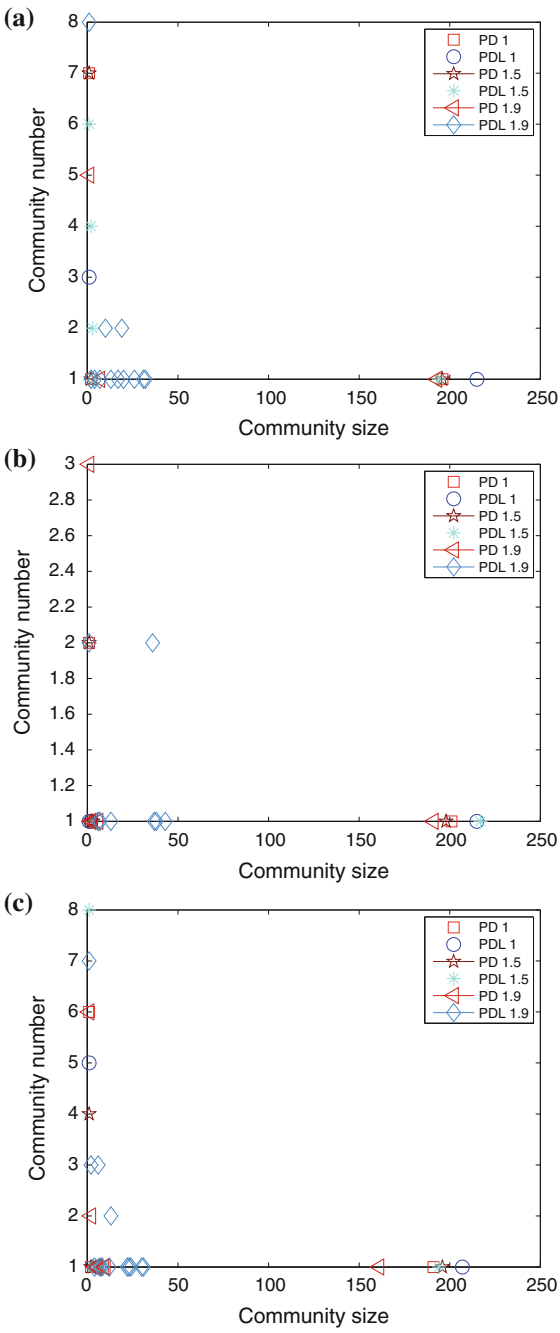

Fig. 13 Number of communities in which the two main parties PDL and PD are split and respective size for IV, V, and VI semesters for different values of γ. **a** IV Semester. **b** V Semester. **c** VI Semester

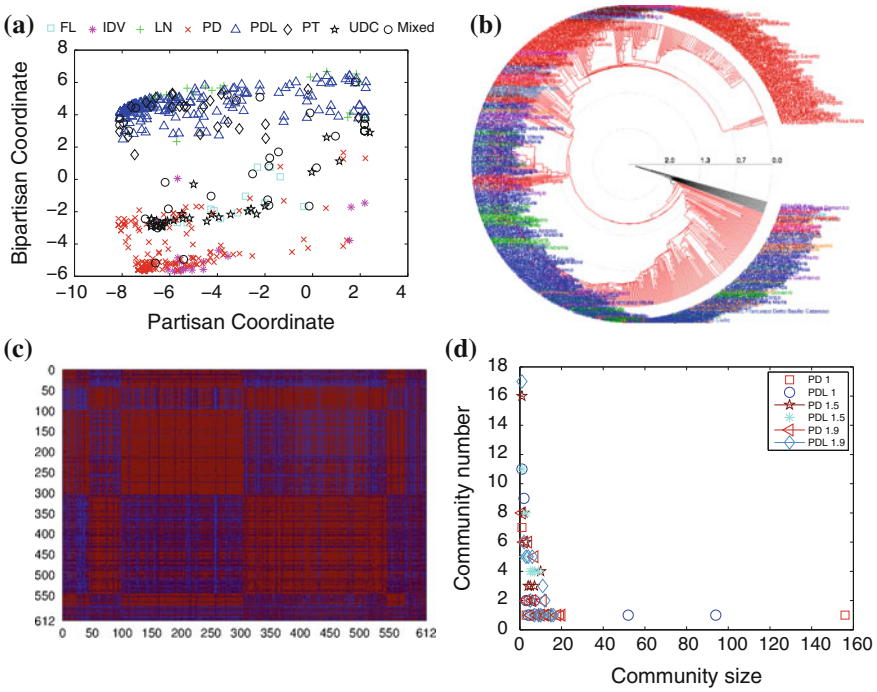

Fig. 14 Results obtained by applying SVD (**a**), hierarchical clustering (**b**), visualization of the similarity matrix (**c**), and community detection (**d**) on the seventh semester

by all the employed methods. First of all, the number of voted measures is less than the fifth part of the other semesters. Furthermore, it happened that the political party organization completely disappeared, and each Parliamentarian voted independently of his group.

Figure 14 gives a clear representation of this situation. In fact, the application of SVD on this semester (Fig. 14a) shows a polarization of all the parties on the first coordinate, and distinguishes between center-left and center-right only on the bipartisan coordinate. Hierarchical clustering returns a unique cluster including all the parties (Fig. 14b), and the visualization of the voting matrices (Fig. 14c) depicts high fragmentation. Finally, Fig. 14d shows that modularity optimization with $\gamma = 1$ extracts a group of 156 and another of 19 members from PD, and two groups of 94 and 52 members from PDL. However, these groups are clustered together, thus confirming the results of the other approaches. For higher values of γ, both parties are split in small groups of at most 20 Parliamentarians, and the communities found are constituted by members of almost all the political parties.

It is worth to note that, as already pointed out, Fig. 11 indicates an abrupt lowering of modularity value in the seventh semester that explains the loss of community structure.

10 Related Work

The investigation of voting records with computational techniques is not new. One of the first paper is that of Jakulin and Buntine [6], where the authors analyzed the United States Senate in the year 2003. They considered the Senate roll calls and the votes cast by each of the US Senators to compute a similarity matrix for every pairs of Senators, based on the Shannon's information theory concept of *mutual information* [14]. The higher the mutual information between two Senators, the greater their similarity. Hierarchical clustering employed on their similarity matrix allowed to distinguish quite precisely between Republicans and Democrats. Furthermore, discrete blocks are identified, and similarity and dissimilarity among these blocks, with the aim of determining the voting influence of a single Senator, computed. The authors observed that, though it is very difficult for a single Senator to influence final voting results because rarely a single vote changes the outcome of a roll call, once the blocks voting in a similar way are detected across a number of roll calls, the influence of changed behavior of a group can be analyzed. In particular, two kinds of altered behavior have been considered: block abstention and block elimination. By using this approach, it was possible to obtain a list of roll calls for which it is deemed that the behavior of a block can affect the outcome.

The same authors, with Pajala [10], analyzed the Finnish Parliament in the year 2003. The Finnish Parliament is composed of 200 members elected for a four-year term. In 2003 elections, changed the cabinet composition, thus Pajala et al. studied the cohesion of new political groups by computing the *agreement index* [4]. They found that the groups composing the majority were more cohesive than the opposition groups. Moreover, they considered the roll calls and the votes cast by each of the Parliamentarians to compute a dissimilarity matrix between every pairs of Parliamentarians, based on Rajski's distance [13], that uses *mutual information* and joint entropy. The lower the Rajski's distance between two Parliamentarians, the greater their similarity. They used the agglomerative hierarchical clustering algorithm *Agnes* [7] with the average linkage method and built dendrograms. All the Parliamentarians were partitioned into clusters by the hierarchical clustering method. The results obtained showed that the analysis performed is able to capture the main characteristics of the Finnish Parliament.

Another interesting study regarding the United States House of Representatives from 101st to 108th Congresses has been done by Porter et al. [12]. They defined bipartite collaboration networks from the assignments of Representatives to House committees and subcommittees. Each edge in the network between two (sub)committees has a weight which corresponds to the normalized interlock. The interlock between two committees is equal to the number of their common members. The normalization is obtained by considering the committee sizes, and dividing the interlock by the expected number of common members, if assignments were defined independently and uniformly at random. Then the hierarchical and modular structure of these networks, by using different community detection methods, has been investigated. Various methods of hierarchical clustering have also been executed. From the

analysis, four hierarchical levels of clustering have been extracted: subcommittees, standing and select committees, groups of standing and select committees, and the entire House. The dendrograms revealed also an organization corresponding to groups of subcommittees inside larger standing committees. In order to perform an analysis of the obtained hierarchies in the House committee networks, authors used the modularity concept, modified to mine committee weighted networks. Instead of counting numbers of edges falling between particular groups, they counted the sums of the weights of those edges. This concept of modularity measures when a particular division of the network has more edge weight within groups than one would expect on the basis of chance, and it is used to evaluate the efficacy of the organizational grouping of the networks and to compare the dendrograms to each other. The community structure of the network of committees has been explored by using three other methods: two based on betweenness values computed on the full bipartite networks of Representatives and committees, and a local community detection algorithm for weighted networks. In this way, the authors identified connections between committees and correlations among committee assignments and Representatives' political positions. Changes in the network structure corresponded to change of Senate majority from Democrats to Republicans. Finally, they applied SVD to evaluate the House roll call votes. From this analysis, it was possible to observe as Democrats are grouped together, and are almost completely separated from Republicans.

Zhang et al. [20] studied the United States Congress by building bipartite networks for Members of Congress. In these "bipartite" networks, there are two types of nodes: Congressman and bills, and a Member of Congress is linked by an edge to each sponsored or cosponsored bill. By using information about the Congressional committee and subcommittee assignments, the authors created another kind of bipartite network where nodes are Representatives and committees/subcommittees, and an edge (i, j) indicates the assignment of Representative i to committee or subcommittee j. Each network is recursively partitioned in order to generate trees or dendrograms to assess its hierarchical structure. This process is able to discover communities of various sizes by iteratively clustering the legislators by using the partitioning algorithm. Modularity evaluates the number of intracommunity versus intercommunity links for a given partition, consequently it has been adopted to quantify the growth in polarization in the U.S. Congress. In particular, during the considered period of 24 years, from the 96th to 108th Congresses, an increase in modularity has been obtained. This corresponded to an increase in party polarization of the Congress that caused the control by the Republicans of both chambers. Authors used also a multidimensional scaling technique called NOMINATE and singular value decomposition analysis. They showed that a matrix of roll call votes can be approximated by using two coordinates: a generic liberal–conservative dimension and a second social dimension. However, the same approaches demonstrated that multiple dimensions are needed to adequately approximate a matrix of cosponsorships. The adopted eigenvector methods detected large communities corresponding to known political cliques. It has been showed that Members of Congress with similar ideologies are clustered together in the identified communities.

Waugh et al. [19] evaluated the polarization in the United States Congress by using also the concept of network modularity. Each node represents a legislator in the network and each edge is the level of agreement between two legislators in roll call voting, indicating the average number of equal votes between them. Generally in a legislature, groups like parties contain strong connections between legislators within the same group but relatively weak connections between individuals in different groups. Multiple community detection algorithms have been employed on the similarity matrices of legislators to identify groups that maximize the modularity inside each roll call network for both the Senate and the House of Representatives. Modularity is adopted for measuring the degree of polarization revealing the main political groups and the divisions among them. A nonmonotonic relationship between maximum modularity and a consequent majority party switch has been explored, demonstrating that the changes in majority are more likely when the modularity value is moderate, uncommon otherwise. In particular, modularity values in Congress t are used to predict modifications in the majority party for Congress $t + 1$. A nonmonotonic relationship between modularity and the stability of the majority party was found in both chambers of Congress. When modularity is low, a change in majority control seems to be less likely; at high levels of modularity, the minority cannot overcome majority cohesion. In both of these cases, it is infrequent to have majority party switches. However, when modularity exhibits medium values, this corresponds to changes taking place for majority cohesion and to a less stable party system.This is called "partial polarization" hypothesis.

At the individual level, some measures associated with modularity, called "divisiveness" and "solidarity" are computed to predict the reelection success for individual House members. The divisiveness measures the effect that each legislator could have on the aggregate polarization of his legislature by using roll call adjacency matrices. About solidity, when its value is close to 1, the legislator and community are strongly aligned. Performing this kind of analysis, authors found that divisiveness has a negative influence on reelection chances and that group solidarity has a positive influence. Furthermore, divisiveness is associated with decreased reelection probability, and the combination of divisiveness and solidarity has a significant positive impact on reelection.

Macon et al. [8] investigated the community structure of networks constructed from voting records of the United Nations General Assembly (UNGA). The UNGA was founded in 1946. Annual sessions from 1946 to 2008 have been considered and unanimous votes removed from the data, because they do not give information about the network structure of voting agreements and disagreements between countries. Three different networks have been defined. The first one is a weighted unsigned network of voting similarities, whose nodes are the countries and whose edges between pairs of countries are weighted by using an agreement measure. This represents the number of agreements on resolutions (yes–yes, no–no, or abstain–abstain) between the two involved countries. The second kind of network is constructed by considering also the number of yes–no disagreements in the elements of the voting similarity matrix. The last kind of network is a signed bipartite network of countries voting for individual resolutions. By analyzing the resolutions with respect to the voting

agreement, the authors were able to detect historical trends and changes in the United General Assembly community structure. In fact, observations appear to be consistent with the expected East–West split of the Cold War and the North–South division of recent sessions that has been detected by social scientists using qualitative methods.

11 Conclusions

The paper presented an investigation of the voting behavior of Italian Parliament in the last years by employing different computational tools. Though studies of this kind exist for different political institutions from US and Europe, as far as we know, this is the first tentative of exploring Italian Parliament with data mining and network analysis methods. We generated networks among the Parliamentarians at consecutive time periods and investigated community structure at multiple scales. By delving the voting records of Representatives, we were capable of characterizing the organizational structure of Parliament, and to discover latent information contained. All the methods used showed to be effective at identifying political parties, and at providing insights on the temporal evolution of groups and their cohesiveness. Future work aims at applying overlapping community detection methods to better uncover hidden collaborations among Parliamentarians of different political membership.

References

1. Brazill Timothy J, Grofman Bernard (2002) Factor analysis versus multi-dimensional scaling: binary choice roll-call voting and the us supreme court. Soc Netw 24(3):201–229
2. Fortunato S, Barthélemy M (2007) Resolution limit in community detection. Proc Natl Acad Sci, USA 104(36)
3. Granell C, Gómez S, Arenas A (2012) Unsupervised clustering analisys : a multiscale complex network approach. J Bifurcat Chaos (in press)
4. Hix S, Noury A, Roland G (2005) Power to the parties: Cohesion and competition in the European parliament. Br J Polit Sci 35(2):209–234
5. Jain AK, Cg Dubes R (1988) Algorithms for clustering data. Prentice Hall, Englewood Cliffs
6. Jakulin A, Buntine W (2004) Analyzing the US senate in 2003: similarities, networks, clusters and blocs. http://eprints.fri.uni-lj.si/archive/00000146/
7. Kauffman L, Rousseeuw PJ (1990) Finding groups in data. Wiley, New York
8. Macon KT, Mucha PJ, Porter MA (2012) Community structure in united nations general assembly. Phys A: Stat Mech Appl 391(1–2):343–361
9. Newman MEJ, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E69:026113
10. Pajala A, Jakulin A, Buntine W (2004) Parliamentary group and individual voting behavior in finnish parliament in year 2003: a group cohesion and voting similarity analysis. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.103.2295
11. Pizzuti C (2012) Boosting the detection of modular community structure with genetic algorithms and local search. In: Proceedings of the 27th symposium on applied computing (SAC 2012), pp 226–231

12. Porter MA, Mucha PJ, Newman MEJ, Friend AJ (2007) Community structure in the United States house of representatives. Phys A: Stat Mech Appl 386(1):414–438
13. Rajiski C (1991) Community structure in congressional cosponsorship networks. Inf Control 4:373–377
14. Shannon C (1948) A mathematical theory of computation. Bell Syst Tech J 27:623–656
15. Strang G (2005) Linear algebra and its applications, IV edn. Thomson Brooks/Cole, Belmont
16. Tan P, Steinbach M, Kumar V (2006) Introduction to data mining. Pearson International Edition, Boston
17. Wasserman S, Faust K (2009) Social network analysis—methods and applications. Cambridge University Press, Cambridge
18. Watts DJ, Strongatz SH (1998) Collective dynamics of 'small-world' networks. Nature 6684(393):440–442
19. Waugh AS, Pei L, Fowler JH, Mucha PJ, Porter MA (2009) Party polarization in congress: a network science approach. http://arxiv.org/abs/0907.3509
20. Zhang Y, Friend AJ, Traud AL, Porter MA, Fowler JH, Mucha PJ (2008) Community structure in Congressional cosponsorship networks. Phys A: Stat Mech Appl 387(7):1705–1712

Glossary

Adjacency matrix The adjacency matrix is a matrix whose rows and columns represent the graph vertices. A matrix entry at position (i, j) contains a 1 or a 0 value according to whether an edge is present between the nodes i and j

Adjective Orientation Similarity The adjective orientation similarity evaluates the semantic orientation similarity of all the adjective terms between two given sentences

Aspect coverage Aspect coverage can be defined as the percentage of topic aspects covered by the summary of reviews

Bipartite networks A bipartite network is a set of network nodes divided into two disjoint sets such that no links are present between two nodes within the same set

CAO An affect analysis system for emoticons created by Michal Ptaszynski

Collaborative filtering collaborative filtering is one common algorithm used for building recommender systems. It tries to predict the utility of an item for a particular user based on the ratings on this item given by other similar users, or the ratings on similar items given by this user; the former is called user-based collaborative filtering, and the latter is called item-based collaborative filtering. (Toward the next generation of recommender systems: a survey of the state of the art and possible extensions, by G. Adomavicius, and A. Tuzhilin, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, Issue 6, pp. 734–749, June 2005)

Complete graph A complete graph is a graph where each pair of vertices is linked by an edge

Computer-mediated communication A way of communication between humans that occurs through the use of two or more electronic devices

Confusion matrices Given a classification model, the confusion matrix indicates in which way the predictions are performed by the model. The rows represent the known classes of the data, i.e., the class labels. The columns are the classes predicted by the model. The value of a matrix entry at position (i, j) corresponds to the number of data items with known class i and predicted class j

Connected components The connected components of a graph represent the set of largest subgraphs, where any two vertices are linked to each other by paths, and which are not connected to other vertices in the original graph

Ö. Ulusoy et al. (eds.), *Recommendation and Search in Social Networks*,
Lecture Notes in Social Networks, DOI 10.1007/978-3-319-14379-8

Connectivity Value Connectivity value structurally represents a user and its connection in the network

Crawling Model Conmponents and their relationships for systematically browsing the World Wide Web (Twitterverse in this case)

Dynamic Network Dynamic network is an architecturally variable network that conforms to the change in the attributes of the entities in the network

Edge weight Edge-weight is the measure of the strength of the relation between the users at either end of the edge. It is computed as a weighted sum of the communication and recommendation flowing through that edge

Edge-source The end of the edge or the link that initiates the communication is called as the source

Edge-target The end of the edge or the link that listens to the communication is called as the target

Emoticons Facial marks or movements that are composed of letter and used in text messages

Evolutionary Principle Incorporating temporal changes by being true to the present and not deviating dramatically from the past

Geodesic A geodesic of a graph G is a shortest path between two vertices (u, v) of G. The length of the maximum geodesic in G is the graph diameter, the length of the minimum geodesic is defined as the graph radius

Glastonbury festival A five-day music festival that takes place near Pilton, Somerset, England

Growth rate It is the ratio of the number of users influenced in a time window to the number of non-influenced users in that time window

Influence flow Influence flow is the spread of influence through the edges/links in the network

Influence value Influence value is computed as a function of the number of influenced neighbors and the strength of the relation between them

Influence Influence is the state where a user starts using a product or service because of direct or viral marketing

Intrinsic Value A normalized score calculated as a composition of various attributes of a user in reference to the marketable product or service

Jaccard similarity Jaccard similarity coefficient measures similarity between two finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets. (Wikipedia)

Joint entropy The joint entropy of two discrete random variables X and Y, with joint probability mass function $p(x, y)$ is defined as

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) log\ p(x, y).$$

Kinesics An interpretation of body motion communication such as facial expressions and gestures

Live edge An edge in a network is a live edge if the source of the edge is influenced and the target is not

Maximal complete subgraphs A maximal complete subgraph of a graph G is a complete subgraph of G which is not properly included in another complete subgraph of G

Maximal frequent sharing patterns A frequent sharing pattern without a proper superset that is frequent

ML-Ask An affect analysis system of textual input in Japanese based on a linguistic assumption that emotional states of a speaker are conveyed by emotional expressions used in emotive utterances. The system was created by Michal Ptaszynski

Music recommender system Music recommender system recommends music to users based on their preferences, interests, or other related information, the commonly used algorithms include content-based, collaborative filtering, and hybrid. (Toward the next generation of recommender systems a survey of the state of the art and possible extensions, by G. Adomavicius, and A. Tuzhilin, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, Issue 6, pp. 734–749, June 2005)

Mutual information The mutual information is the uncertainty reduction in one random variable given the knowledge about the other one. If the mutual information is high, it represents a huge reduction in uncertainty; if the mutual information is low, it indicates a small reduction; if the mutual information between the two random variables is zero, it means that the variables are independent. Given two discrete variables X and Y with joint probability distribution $P_{X,Y}(x, y)$, the mutual information between them is $I(X; Y) = \sum_{x,y} P_X Y(x, y) log \frac{P_{XY}(x,y)}{P_X(x)P_Y(y)} = E_{P_{XY}} log \frac{P_{XY}}{P_X P_Y}$, where $P_X(x)$ and $P_Y(y)$ are the marginals, $P_X(x) = \sum_y P_{XY}(x, y)$ and $P_Y(y) = \sum_x P_{XY}(x, y)$ and E_P is the expected value over the distribution P

Network Value Network value is the measure of a user's capacity as an influencer for a product or service in the network. It is a function of the intrinsic value and the connectivity value of a user for that product or service

Nonuniform Random Walks A random walk whose next hop is not selected uniformly at random out of the available choices

Online Paid Posters Users who get paid to write promotional or fake reviews and comments online

Orthogonal matrix A $n \times n$ matrix A is an orthogonal matrix if $AA^T = I$, where A^T is the transpose of A and I is the identity matrix

Patterns Patterns are consistent and recurring features that help to model a phenomenon or problem, and are useful as indicators or models for predicting its future trend

Pearson correlation A measure of the linear correlation (dependence) between two variables X and Y in statistics

Pearson correlation is a measure of the linear correlation between two variables X and Y, giving a value between +1 and −1 inclusive, where 1 is the total positive correlation, 0 is no correlation, and −1 is total negative correlation. (Wikipedia)

PerSocial Relevance A relevance model that determines the social relevance between a user and a document

PerSocialization Personalization of search results using social signals

Personalized Search Engine (PERSOSE) Search engine that uses social signals to personalize the search results

Polarity distribution preservation Polarity distribution preservation evaluates the correlation of aspect-level polarity between reviews and the system generated summary

Precision In the context of information retrieval, precision is the fraction of the retrieved documents which are relevant. (Modern Information Retrieval: The Concepts and Technology behind Search (2nd Edition), Ricardo Baeza-Yates, and Berthier Ribeiro-Neto, Addison Wesley, 2010, ISBN 9780321416919)

Predictive Accuracy Measures how close a predicted value (given by the Recommender System) is to a withheld actual rating

Rate limit An upper limit set by Twitter that used to control the rate of requests per user

Recall In the context of information retrieval, recall is the fraction of the relevant documents which have been retrieved by the information retrieval system. (Modern Information Retrieval The Concepts and Technology behind Search (2nd Edition), Ricardo Baeza-Yates, and Berthier Ribeiro-Neto, Addison Wesley, 2010, ISBN 9780321416919)

Recommendation List Diversity Measures how different the items of a recommendation list are from one another

Recommendation List Novelty Measures the extent to which an item (or a set of items) is new when compared with those items that have already been consumed by a user (or a community of users)

Recommendation score It is an average score of the user and its connections to recommend a product or service in the network

Recommender Systems Software systems that aim to propose new items that have not been evaluated by the users yet

Rejection Sampling A statistical technique for generating samples from a hard-to-sample distribution by employing as an instrument an easy-to-sample distribution

Representatives A representative is an individual who represents a constituency or community in a legislative structure, i.e., a member of the US House of Representatives

Request sharing pattern A combination of vertex labels that is shared within a connected subgraphs with a minimum number of vertices

Roll calls The roll calls are voting processes where legislators are called on by name and have the possibility to cast their vote or to abstain

SE (Search Engine) Search engines are services that crawl very large amount of data (documents or websites, for example) and can efficiently search them for keywords to return a list of matching documents

Shortest paths The shortest path between two vertices i and j is a path such that the sum of the weights of its edges is minimized with respect to the other possible paths between them. For unweighted graphs, every edge is weighted as 1

SMQA (Social Media Question Asking) Often people use social networking sites to ask queries to their network members, or to generic people using that service. Researchers have termed this as social media question asking (SMQA). The social networking site concerned may be of general purpose (e.g., Facebook, Twitter) or provide specific type of service (for example, Jelly)

SNS (Social Networking Sites) Services that enable the users to build and use social connectivity with other users. The concept of social networks predates the computer era. But the widespread penetration of the Internet, especially with proliferation of mobile computing devices has made the implementation of social networking sites a success. Common examples are Facebook, Twitter, Google+, Weibo, etc.

Social Actions Set of actions that a given user can perform on any document. Examples include LIKE, RECOMMEND, and SHARE

Social network A social network is defined as a network of interactions or relationships, where the nodes consist of actors, and the edges consist of the relationships or interactions between these actors. (Social Network Data Analytics, edited by Charu C. Aggarwal, ISBN: 978-1-4419-8461-6, Springer, 2011)

Social tagging Tagging is a process where a user assigns a tag to a web object or resource; social tagging is to tag the object during the social interactions supported by the social networking site. (Social Network Data Analytics, edited by Charu C. Aggarwal, ISBN: 978-1-4419-8461-6, Springer, 2011)

Spam Detection Use machine learning techniques to identify the potential online paid posters

Strongly-Connected group A group of users within a larger network having a strong association within the group than outside is termed as a strongly connected group. The association is represented by the weight of the edges connecting them

Supervised Learning Machine learning task of inferring a function from labeled training data

Temporal update Temporal update is the incorporation of the temporal changes in the network

TF-IDF Short for term frequency-inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus

Training Data Manually labelled samples

Trust Networks Social networks that are comprised of trust statements among the actors

Twitter API Twitter is an online social networking service that enables users to send and read short 140-character messages called "tweets". The Twitter's application programming interface (API) allows other Web services and applications to integrate with Twitter

Twittersphere/Twitterverse The entire Twitter world, especially the postings made on the social media website Twitter, considered collectively

Unsupervised Learning Find hidden structure in unlabeled data

Viral Marketing Viral Marketing is a marketing methodology that relies on getting the customers of a product or service to promote it to their connections in the network

Virtual Network It is a network comprising of digital links between the entities within a network. Entities communicate with each other via these links. The links represent the relation between the entities

Voting records Voting records are lists containing the voting history of candidates or elected officials

Index

Ö. Ulusoy et al. (eds.), *Recommendation and Search in Social Networks*,
Lecture Notes in Social Networks, DOI 10.1007/978-3-319-14379-8